Einführung in Visual Computing

186.822

Wiederholung und Übung

Werner Purgathofer

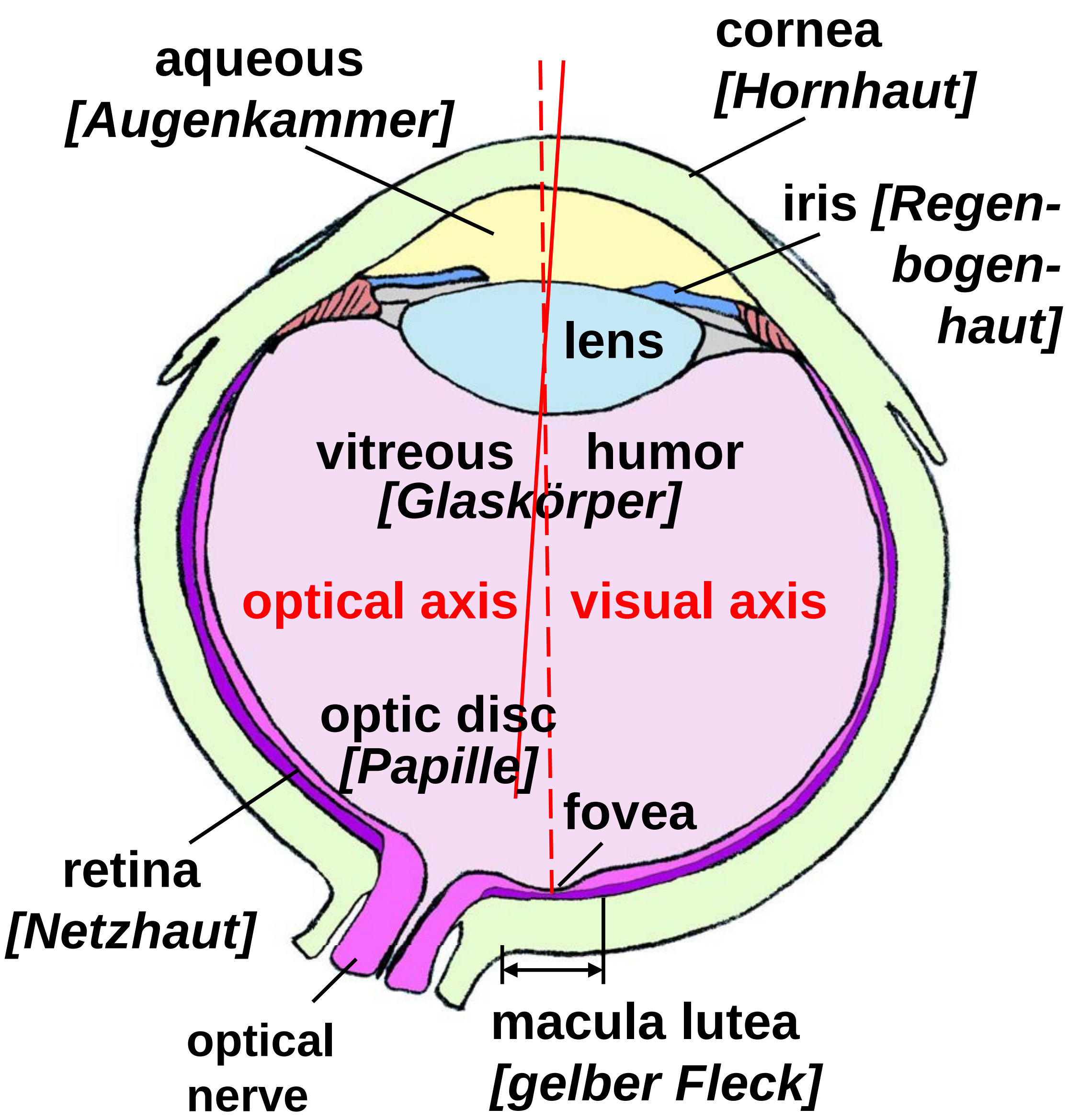
Hinweise zum 1. Test

- Donnerstag 6.4. 18 Uhr
- Hörsaaleinteilung auf Webseite unter “Aktuelles”
- *keine* Unterlagen erlaubt:
  - kein Skriptum, keine Folienkopien
  - kein Computer, Pad, Handy
  - kein programmierbarer Taschenrechner
  - kein Nachbar!
- *einfache* Taschenrechner und *unkommentierte* Formelsammlungen erlaubt
- Bewertung der wahr/falsch-Fragen:
  - Pluspunkte wenn richtig
  - Minuspunkte wenn falsch
  - aber nie weniger als Null pro Fragenblock

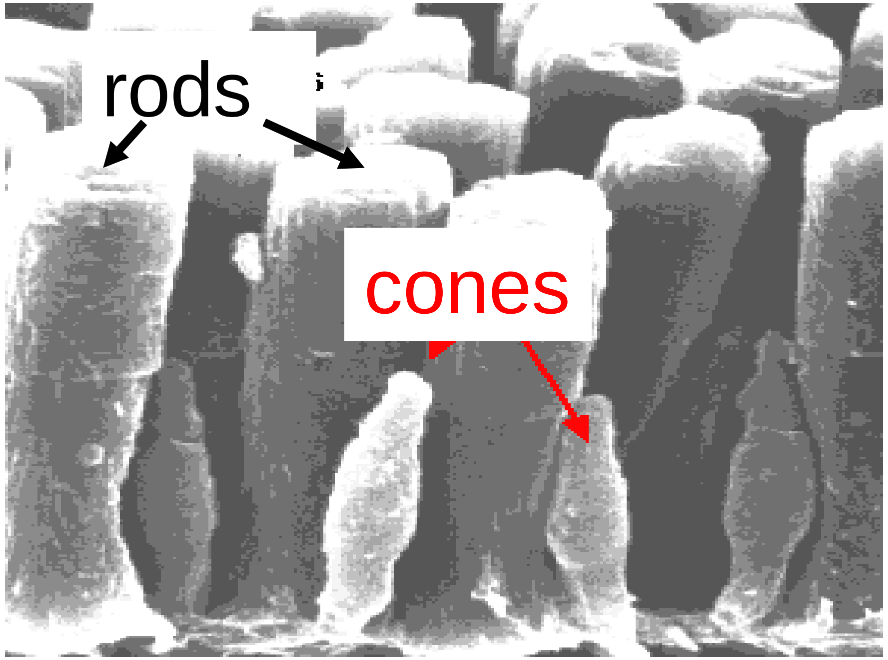
Farben und Farbfehlsichtigkeit

- Die Netzhaut des Auges enthält Stäbchen für das Helligkeits-Sehen und 3 Zapfenarten für das Farbsehen. □ **wahr** □ **falsch**
- Rot-grün-blinde Personen sehen nur die Farben Rot und Grün. □ **wahr** □ **falsch**
- Licht mit höherer Frequenz hat eine kleinere Wellenlänge. □ **wahr** □ **falsch**
- Die Purpurlinie des CIE-Chromaticitydiagramms besteht aus spektralreinen Farben. □ **wahr** □ **falsch**
- Drucker verwenden das __________-Farbmodell.

The Human Eye



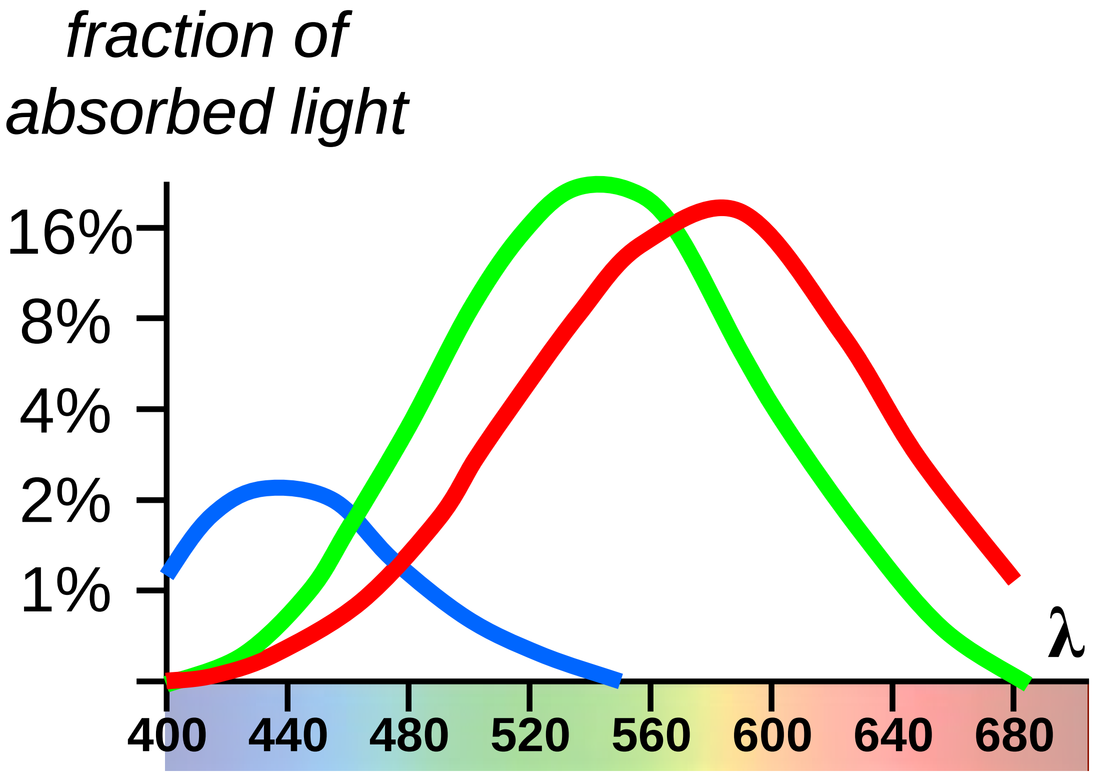
- retina contains
  - rods: b/w
  - cones: color



The Human Eye

- 3 types of cones

- different wavelength sensitivities:
  - red
  - green
  - blue

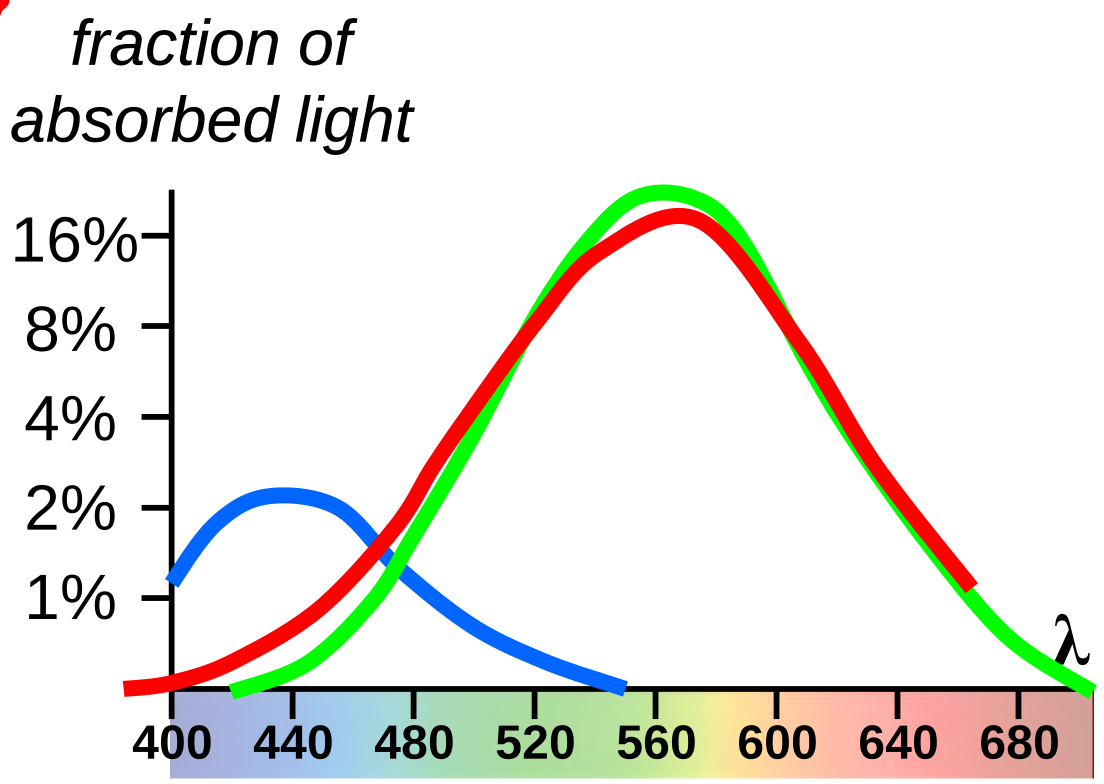


Farben und Farbfehlsichtigkeit

- Die Netzhaut des Auges enthält Stäbchen für das Helligkeits-Sehen und 3 Zapfenarten für das Farbsehen. ☒ **wahr** ☐ **falsch**
- Rot-grün-blinde Personen sehen nur die Farben Rot und Grün. ☐ **wahr** ☐ **falsch**
- Licht mit höherer Frequenz hat eine kleinere Wellenlänge. ☐ **wahr** ☐ **falsch**
- Die Purpurlinie des CIE-Chromaticitydiagramms besteht aus spektralreinen Farben. ☐ **wahr** ☐ **falsch**
- Drucker verwenden das __________-Farbmodell.

Color Blindness

- red/green blindness
  - red & green cones too similar
- blue blindness
  - no blue cones
- monochromatism
  - all cones missing

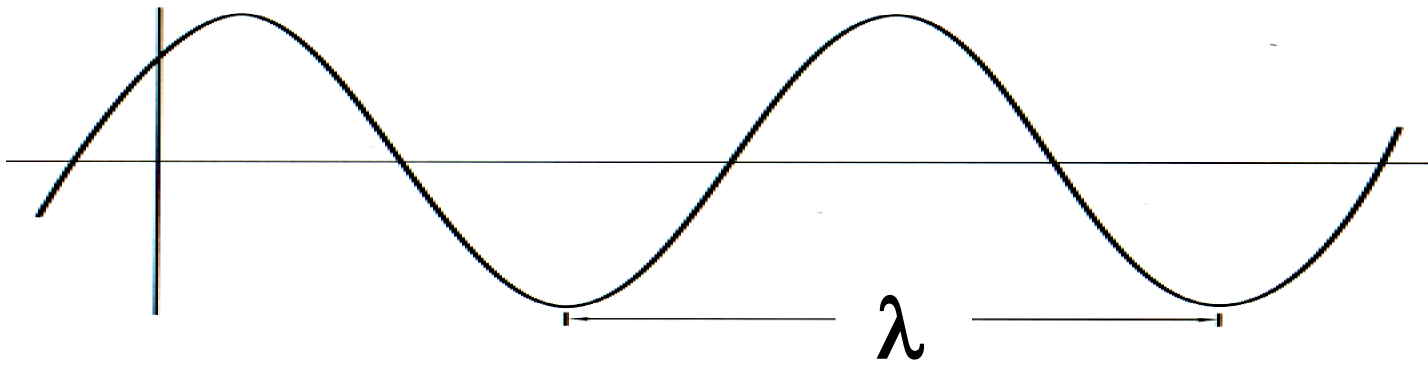


Farben und Farbfehlsichtigkeit

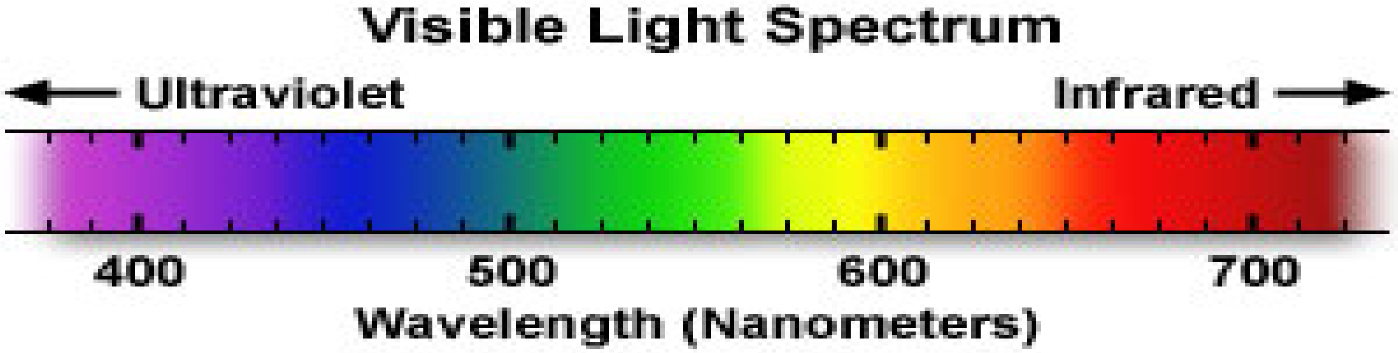
- Die Netzhaut des Auges enthält Stäbchen für das Helligkeits-Sehen und 3 Zapfenarten für das Farbsehen. ☒ **wahr** ☐ **falsch**
- Rot-grün-blinde Personen sehen nur die Farben Rot und Grün. ☐ **wahr** ☒ **falsch**
- Licht mit höherer Frequenz hat eine kleinere Wellenlänge. ☐ **wahr** ☐ **falsch**
- Die Purpurlinie des CIE-Chromaticitydiagramms besteht aus spektralreinen Farben. ☐ **wahr** ☐ **falsch**
- Drucker verwenden das __________-Farbmodell.

Light - An Electromagnetic Wave

- light is electromagnetic energy
- monochrome light can be described either by frequency f or wavelength λ
- $c = \lambda \cdot f$ (c = speed of light)



- shorter wavelength equals higher frequency

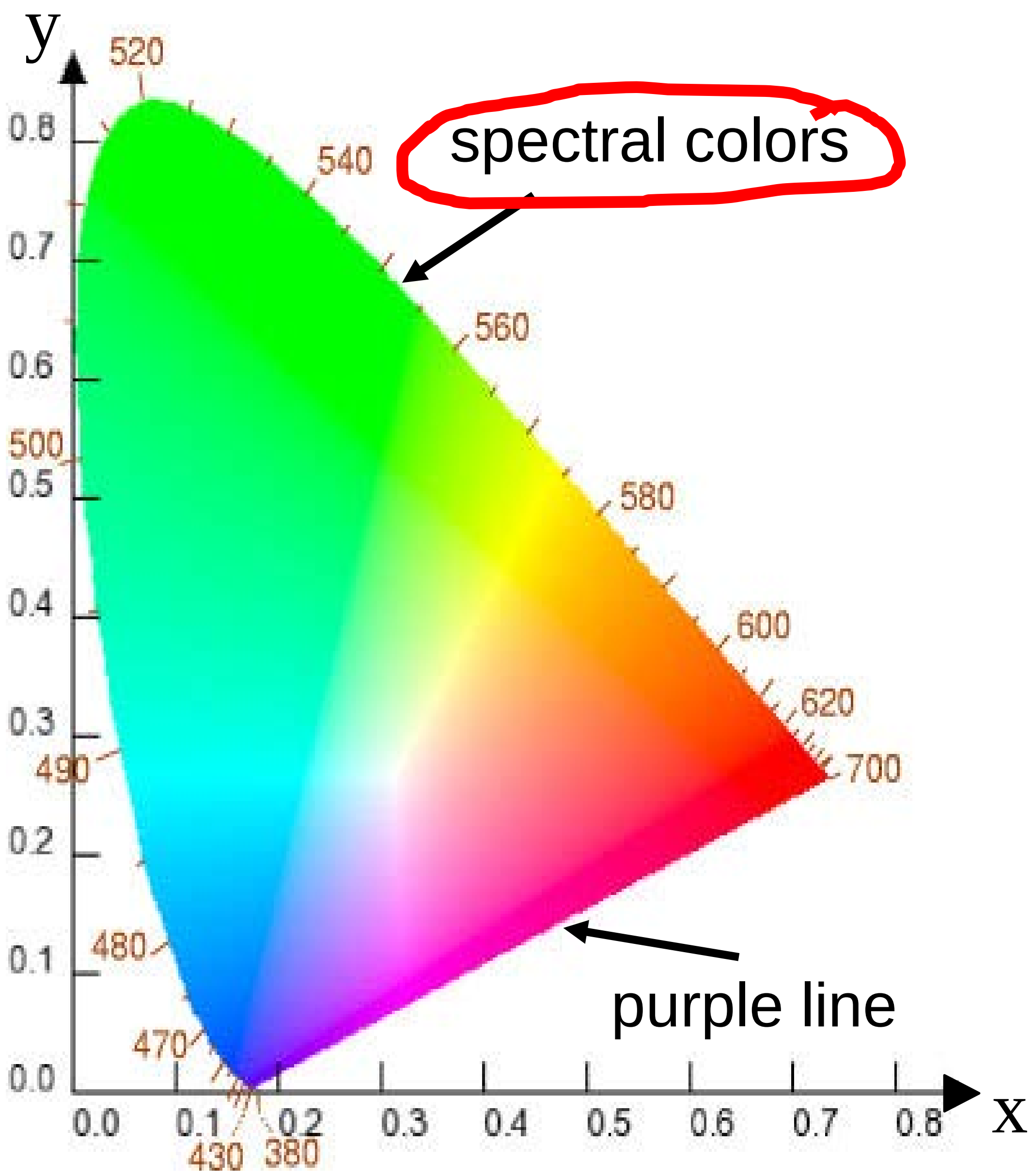


- red ≈ 700 nm
- violet ≈ 400 nm

Farben und Farbfehlsichtigkeit

- Die Netzhaut des Auges enthält Stäbchen für das Helligkeits-Sehen und 3 Zapfenarten für das Farbsehen. ☒ **wahr** □ **falsch**
- Rot-grün-blinde Personen sehen nur die Farben Rot und Grün. □ **wahr** ☒ **falsch**
- Licht mit höherer Frequenz hat eine kürzere Wellenlänge. ☒ **wahr** □ **falsch**
- Die Purpurlinie des CIE-Chromaticitydiagramms besteht aus spektralreinen Farben. □ **wahr** □ **falsch**
- Drucker verwenden das __________-Farbmodell.

CIE Chromaticity Diagram



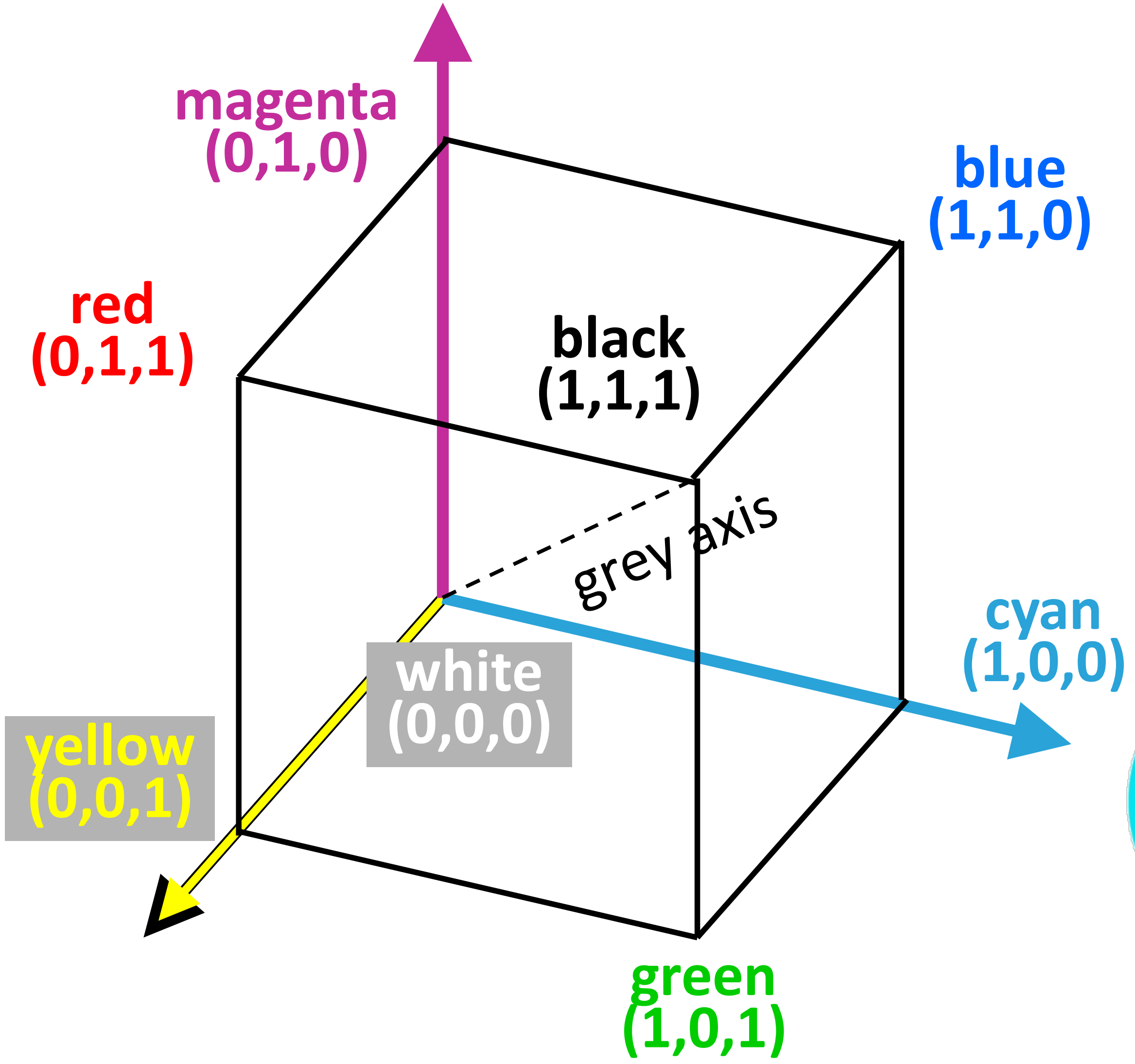
- identifying complementary colors
- determining dominant wavelength & purity
- comparing color gamuts

spectral color positions are along the boundary curve

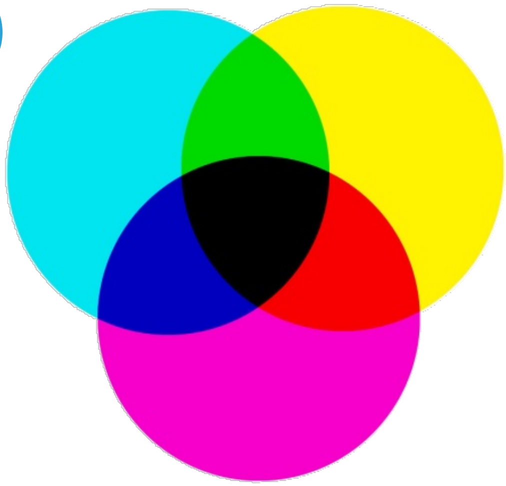
Farben und Farbfehlsichtigkeit

- Die Netzhaut des Auges enthält Stäbchen für das Helligkeits-Sehen und 3 Zapfenarten für das Farbsehen. ☒ **wahr** ☐ **falsch**
- Rot-grün-blinde Personen sehen nur die Farben Rot und Grün. ☐ **wahr** ☒ **falsch**
- Licht mit höherer Frequenz hat eine kürzere Wellenlänge. ☒ **wahr** ☐ **falsch**
- Die Purpurlinie des CIE-Chromaticitydiagramms besteht aus spektralreinen Farben. ☐ **wahr** ☒ **falsch**
- Drucker verwenden das __________-Farbmodell.

CMY Color Model



- primary colors:
  cyan, magenta, yellow
- ***subtractive*** color model
  (for *hardcopy devices*)
  - C = G + B, using C "subtracts" R



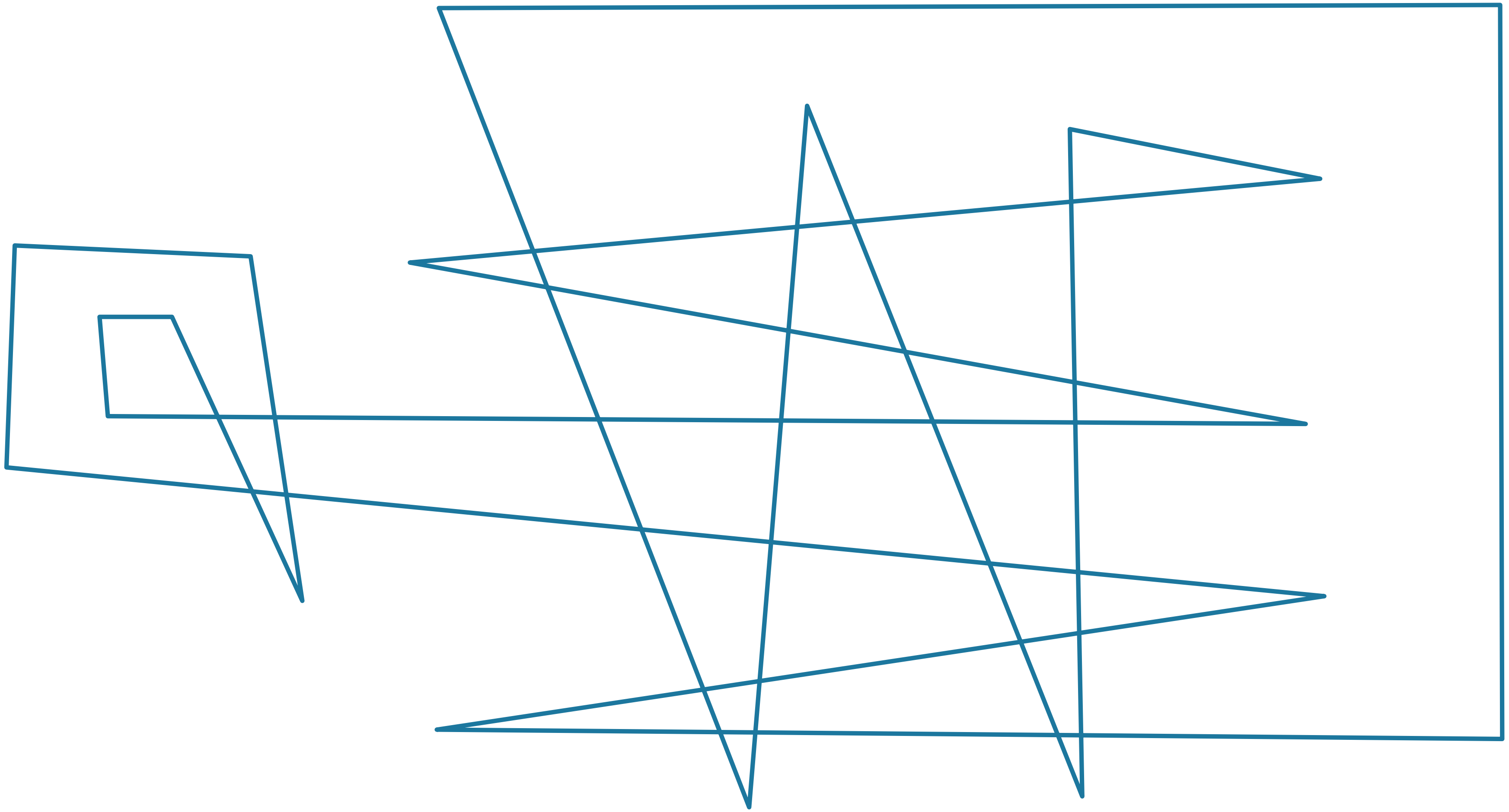
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Farben und Farbfehlsichtigkeit

- Die Netzhaut des Auges enthält Stäbchen für das Helligkeits-Sehen und 3 Zapfenarten für das Farbsehen. ☒ **wahr** ☐ **falsch**
- Rot-grün-blinde Personen sehen nur die Farben Rot und Grün. ☐ **wahr** ☒ **falsch**
- Licht mit höherer Frequenz hat eine kürzere Wellenlänge. ☒ **wahr** ☐ **falsch**
- Die Purpurlinie des CIE-Chromaticitydiagramms besteht aus spektralreinen Farben. ☐ **wahr** ☒ **falsch**
- Drucker verwenden das <u>CMY(K)</u>-Farbmodell.

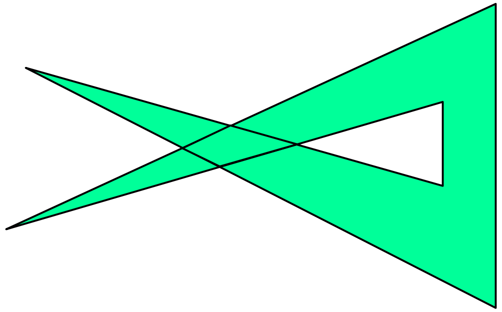
Beispiel: Polygonfüllen

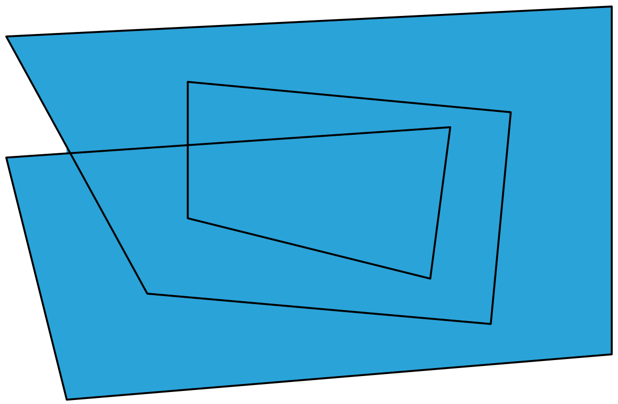
Füllen Sie das Polygon nach der Non-Zero-Winding-Number Regel!

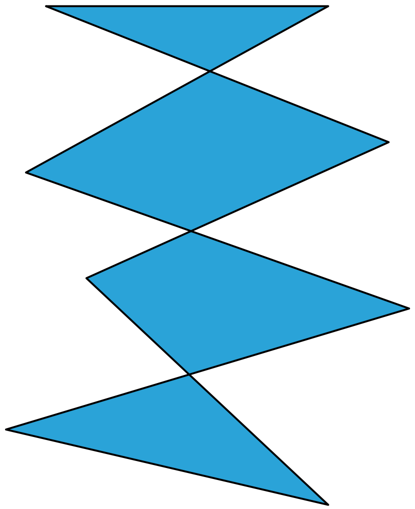


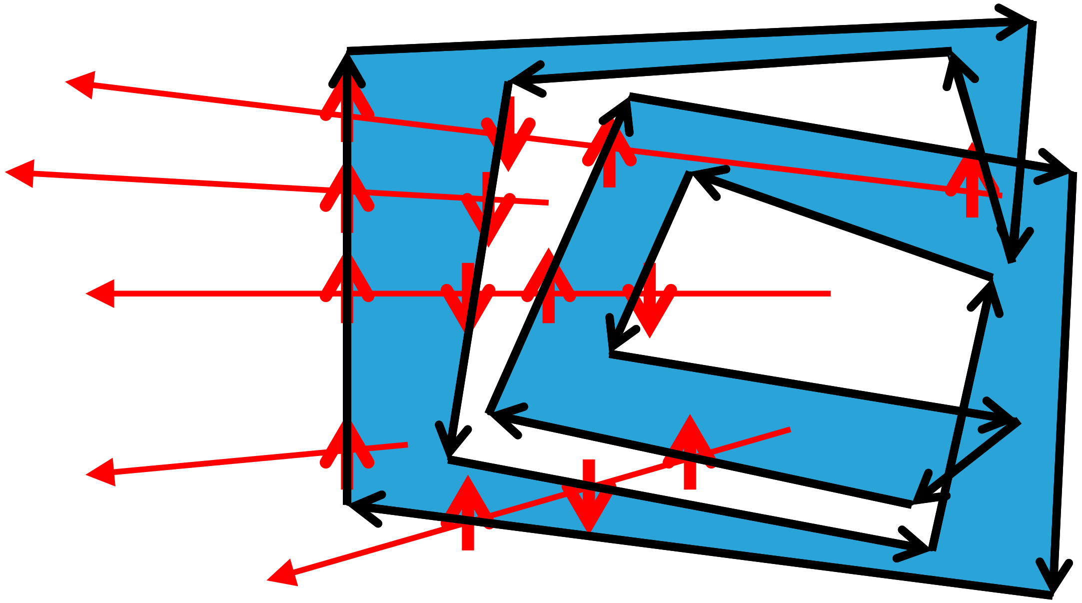
What is Inside?: Nonzero Winding Number

- point is inside if polygon surrounds it
- straight line to the outside:
  - same # edges up and down = outside
  - different # edges up and down = inside



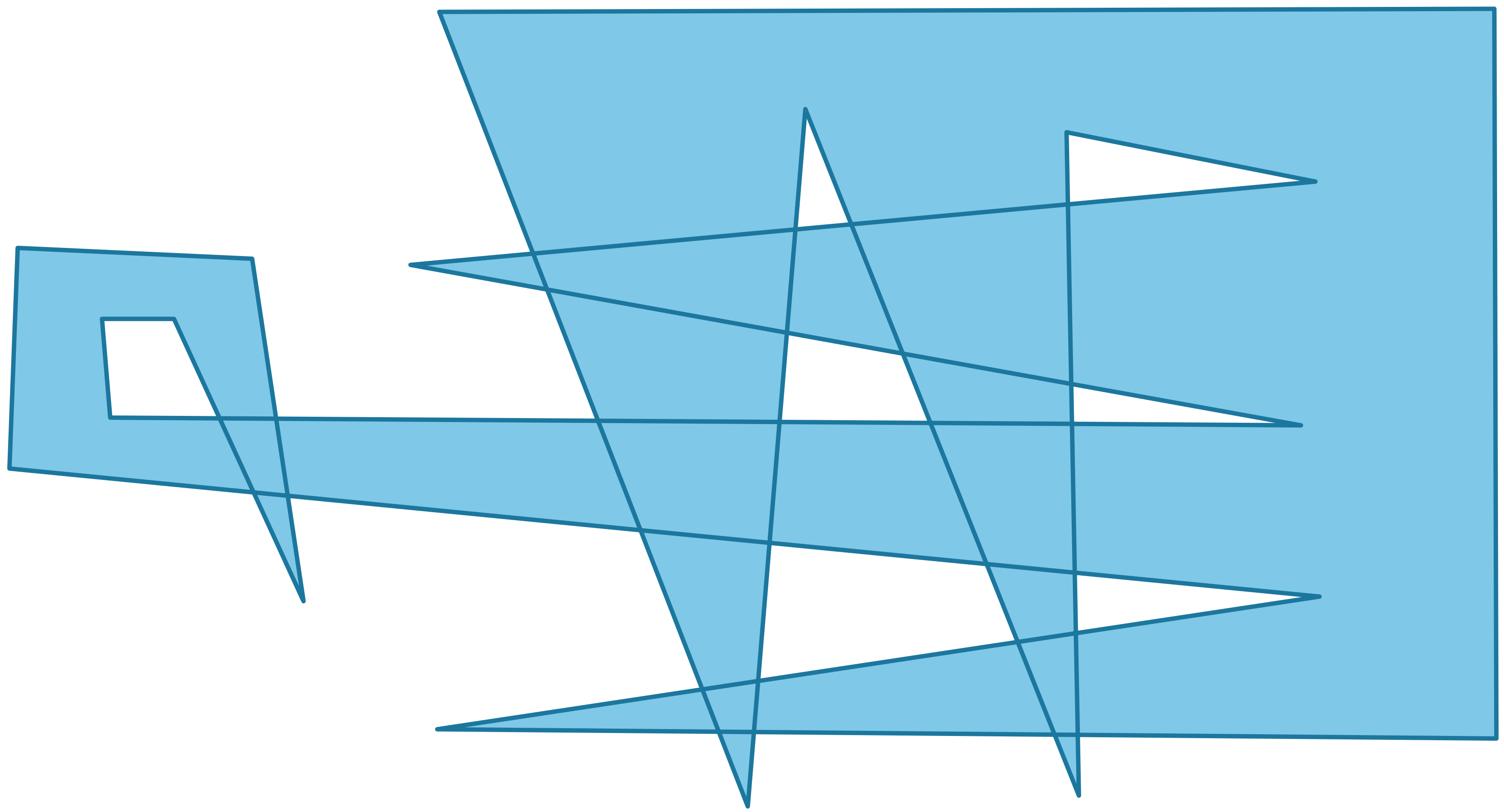






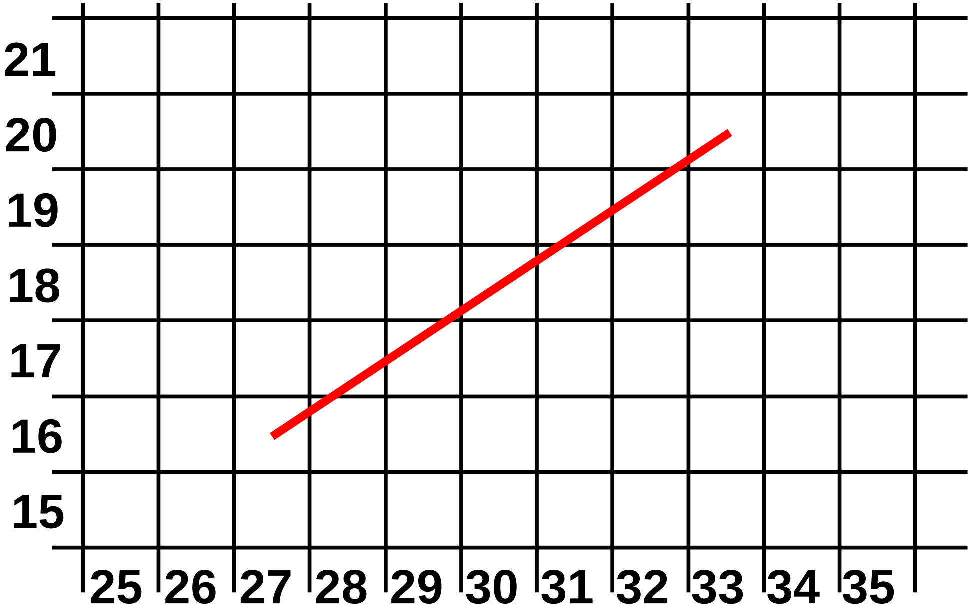
Beispiel: Polygonfüllen

Lösung



Beispiel: Rasterisierung

Zeichnen Sie die vom **Bresenham-Verfahren** erzeugten Pixel ein und geben Sie die Werte der Entscheidungsvariablen an!



$$p_0 = 2\Delta y - \Delta x$$

if $p_k < 0$

then draw pixel (x_k+1, y_k);

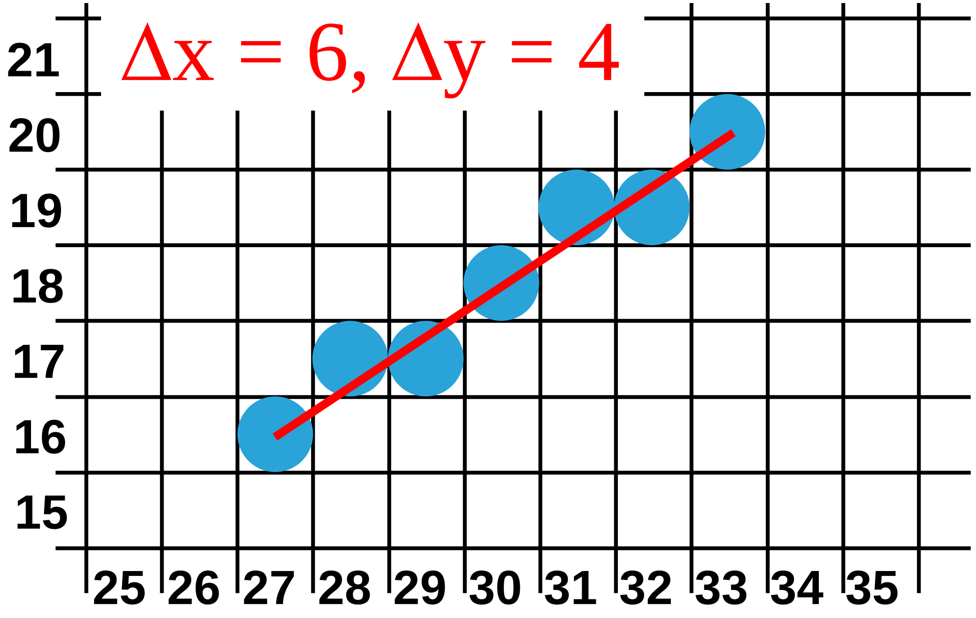
$$p_{k+1} = p_k + 2\Delta y$$

else draw pixel (x_k+1, y_k+1);

$$p_{k+1} = p_k + 2\Delta y - 2\Delta x$$

Bresenham: Example

k	p_k	(x_{k+1},y_{k+1})
		(27,16)
0	2	(28,17)
1	-2	(29,17)
2	6	(30,18)
3	2	(31,19)
4	-2	(32,19)
5	6	(33,20)



$p_0 = 2\Delta y - \Delta x \quad = 2$

if $p_k<0$

then draw pixel (x_k+1,y_k);

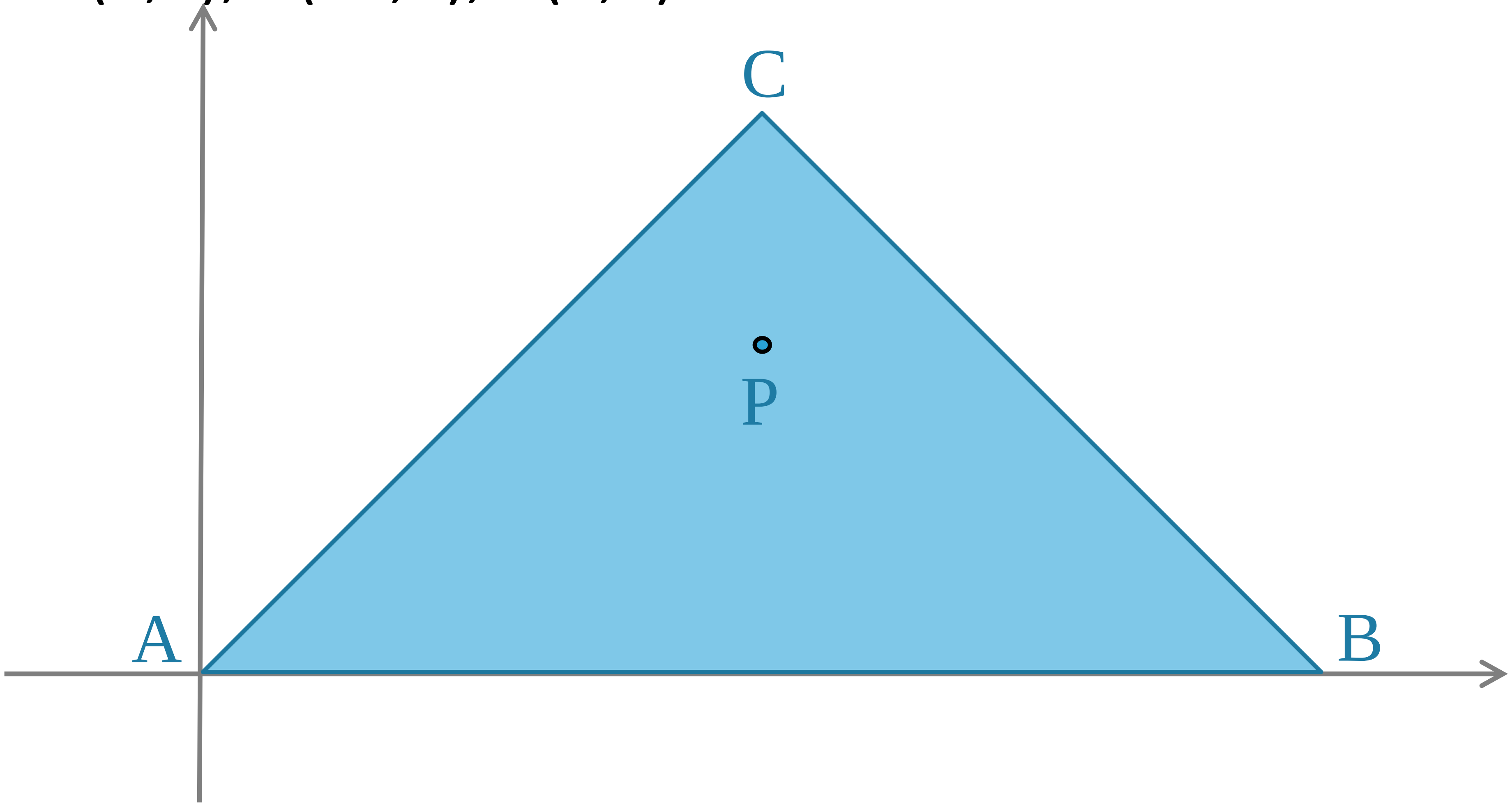
$p_{k+1} = p_k + 2\Delta y \quad + 8$

else draw pixel (x_k+1,y_k+1);

$p_{k+1} = p_k + 2\Delta y - 2\Delta x \quad - 4$

Beispiel: Baryzentrische Koordinaten

Welche baryzentrischen Koordinaten hat der Punkt P(5,3) im Dreieck A(0,0), B(10,0), C(5,5)?

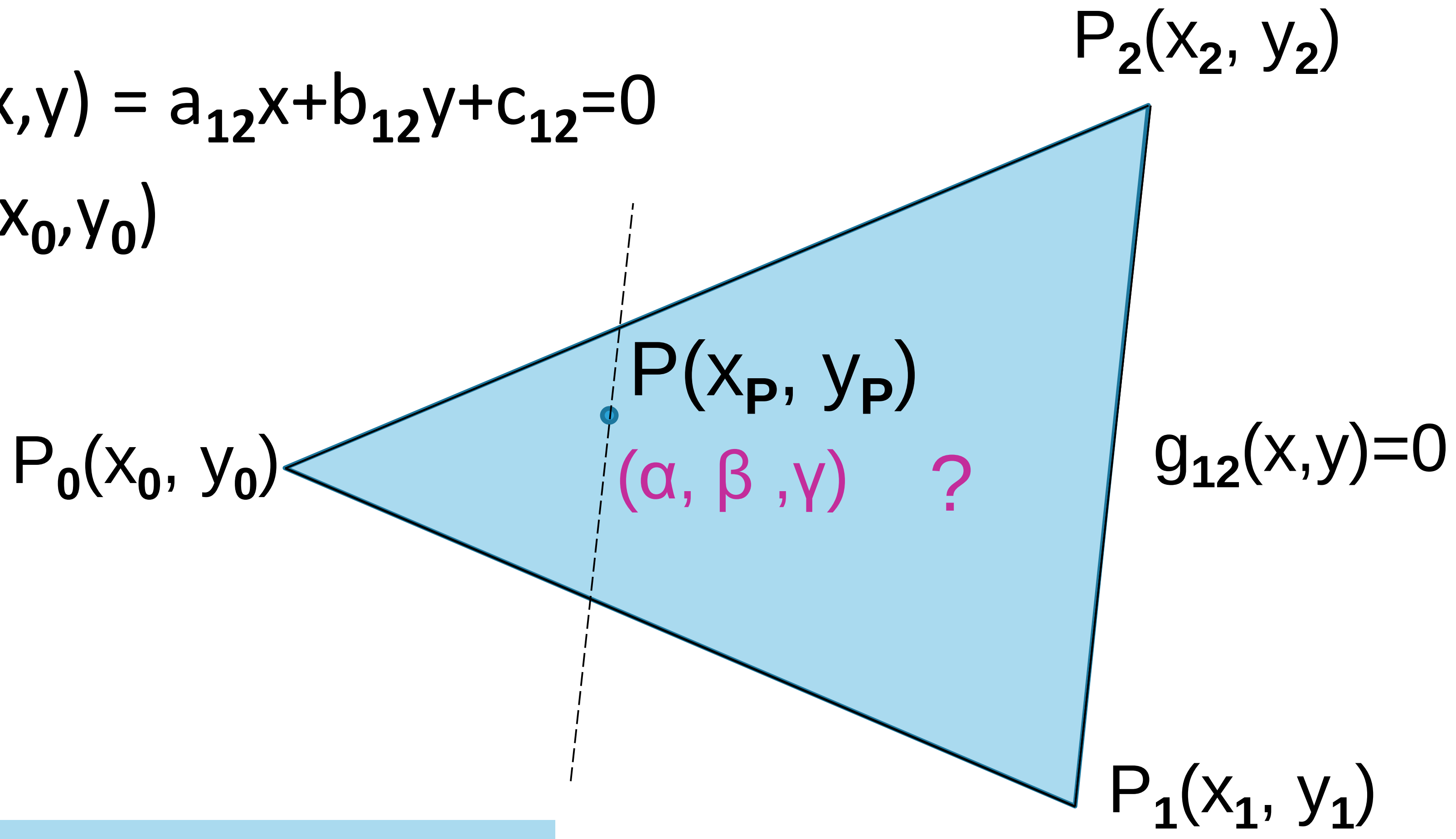


Computing (α, β ,γ) for P(x,y)

line through P_1, P_2: $g_{12}(x,y) = a_{12}x+b_{12}y+c_{12}=0$

then $\alpha = g_{12}(x_P,y_P) / g_{12}(x_0,y_0)$

β,γ analogous

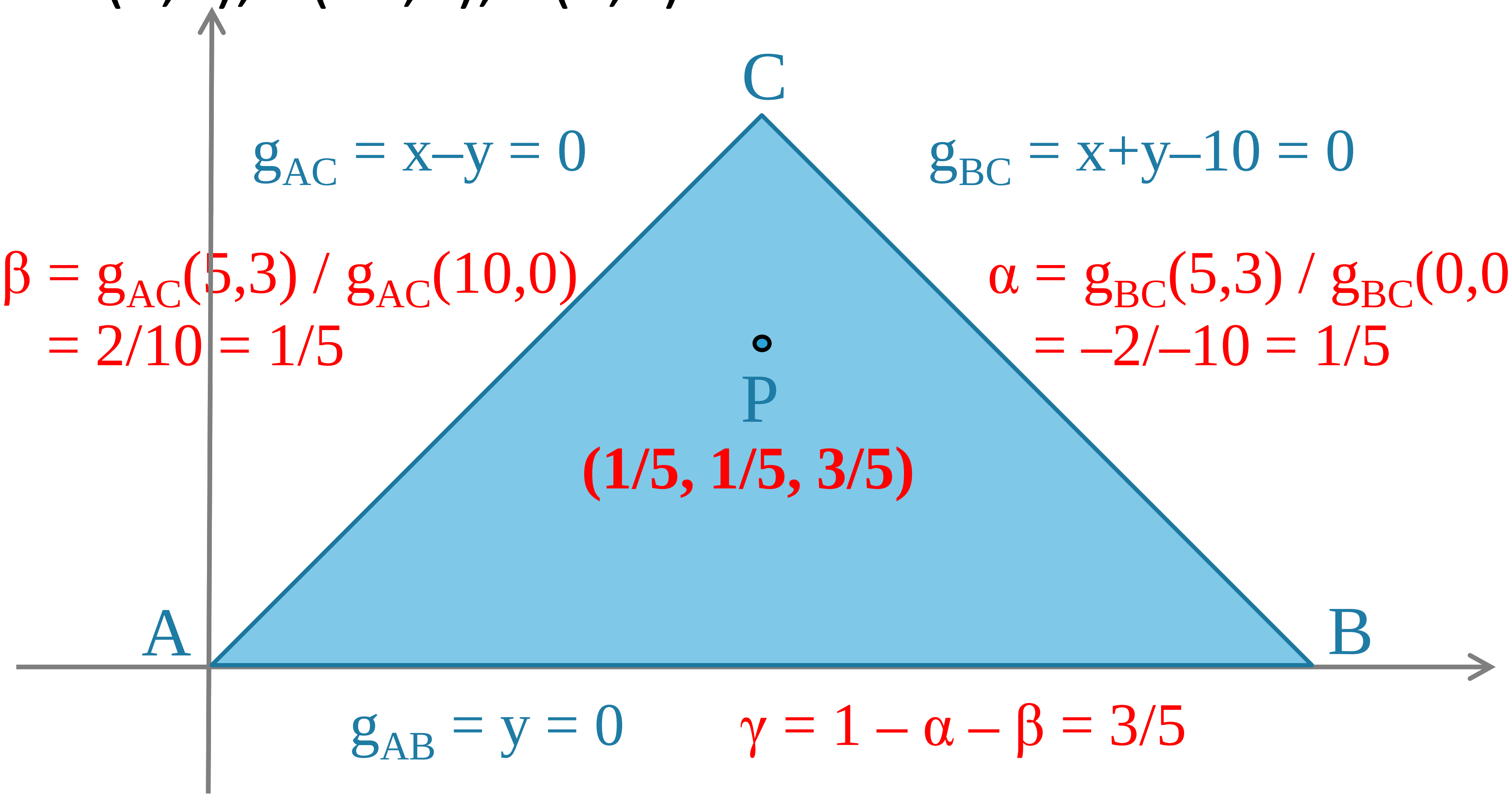


$P = \alpha P_0 + \beta P_1 + \gamma P_2$

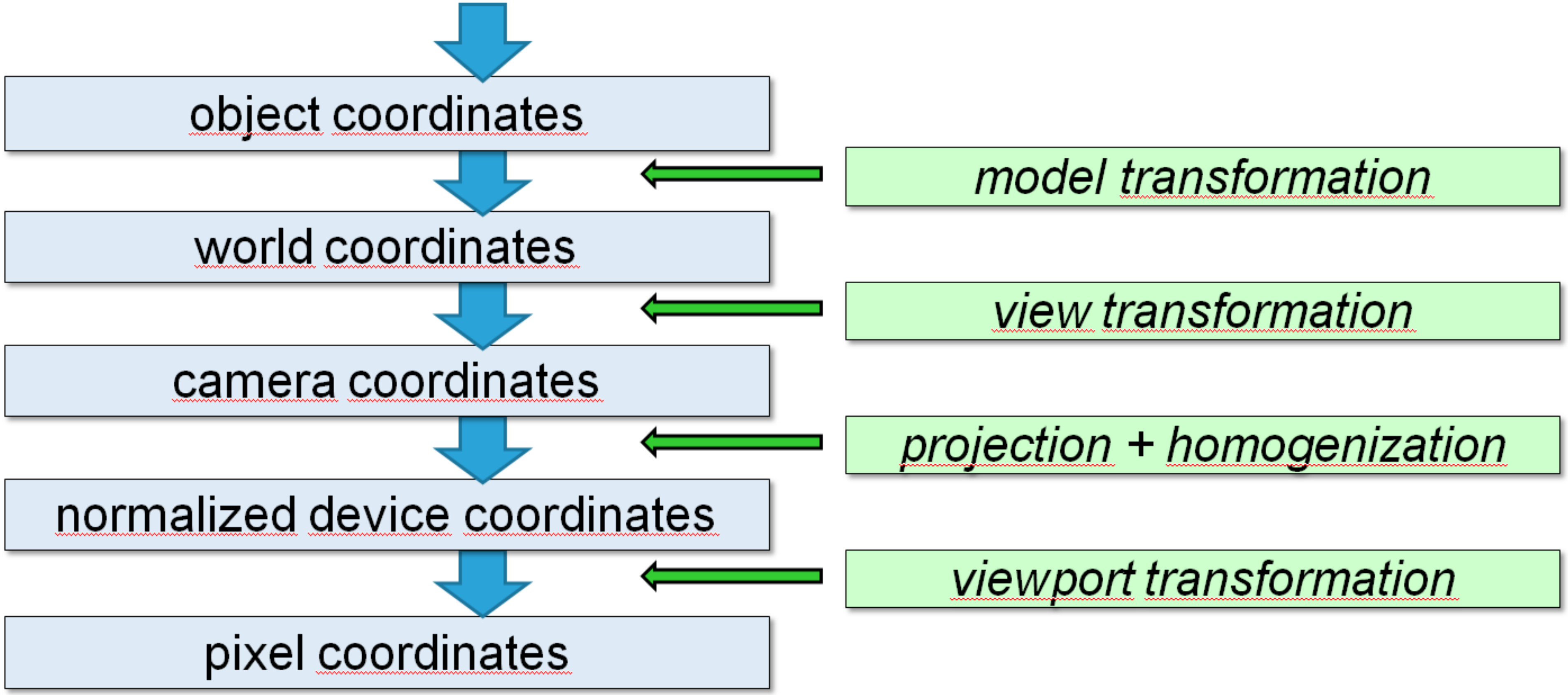
triangle = $\{P \mid \alpha+\beta+\gamma=1,\ 0<\alpha<1,\ 0<\beta<1,\ 0<\gamma<1\}$

Beispiel: Baryzentrische Koordinaten

Welche baryzentrischen Koordinaten hat der Punkt P(5,3) im Dreieck A(0,0), B(10,0), C(5,5)?



- Nach der View-Transformation befindet man sich im
  ☒ Kamera- ☐ Objekt- ☐ Pixelkoordinatensystem



Objektrepräsentation

- Bei einer B-Rep wird nur die Oberfläche der Objekte beschrieben.
  □ **wahr** □ **falsch**

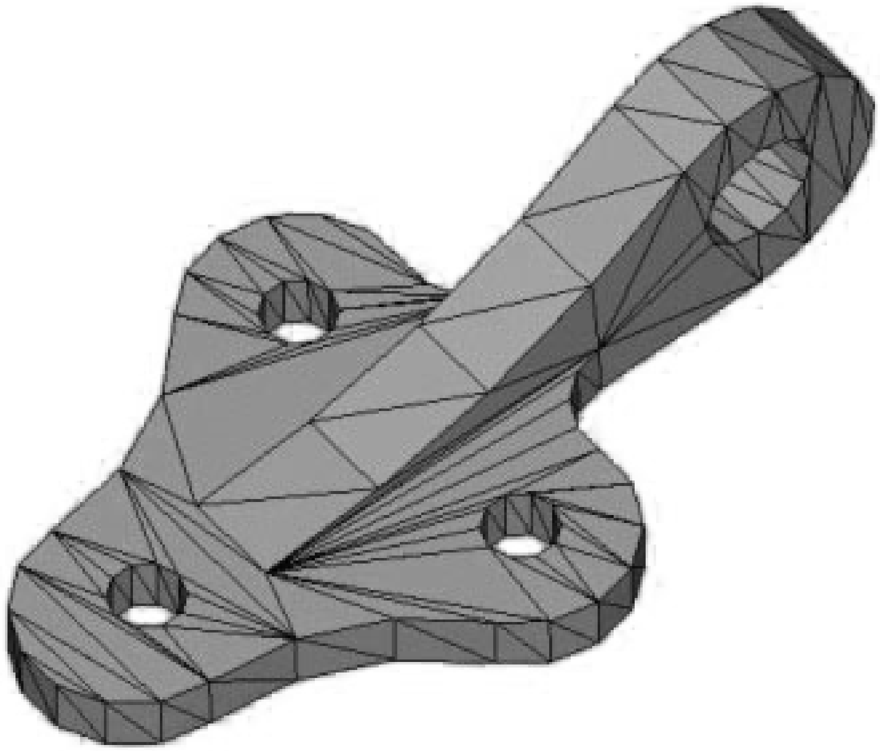
- CSG-Objekte werden durch die Operatoren Vereinigung, Durchschnitt und Mengendifferenz beschrieben. □ **wahr** □ **falsch**

- Der einzige Weg um CSG-Objekte zu zeichnen ist sie in BRep-Objekte umzurechnen. □ **wahr** □ **falsch**

- Ein Szenengraph ist eine genormte Datenstruktur zum Austausch geometrischer Daten. □ **wahr** □ **falsch**

Polygon Surfaces (1)

- set of surface polygons enclose object interior
  = ***Boundary Representation***
  ("B-Rep")



example:
machine part surface
represented by triangles

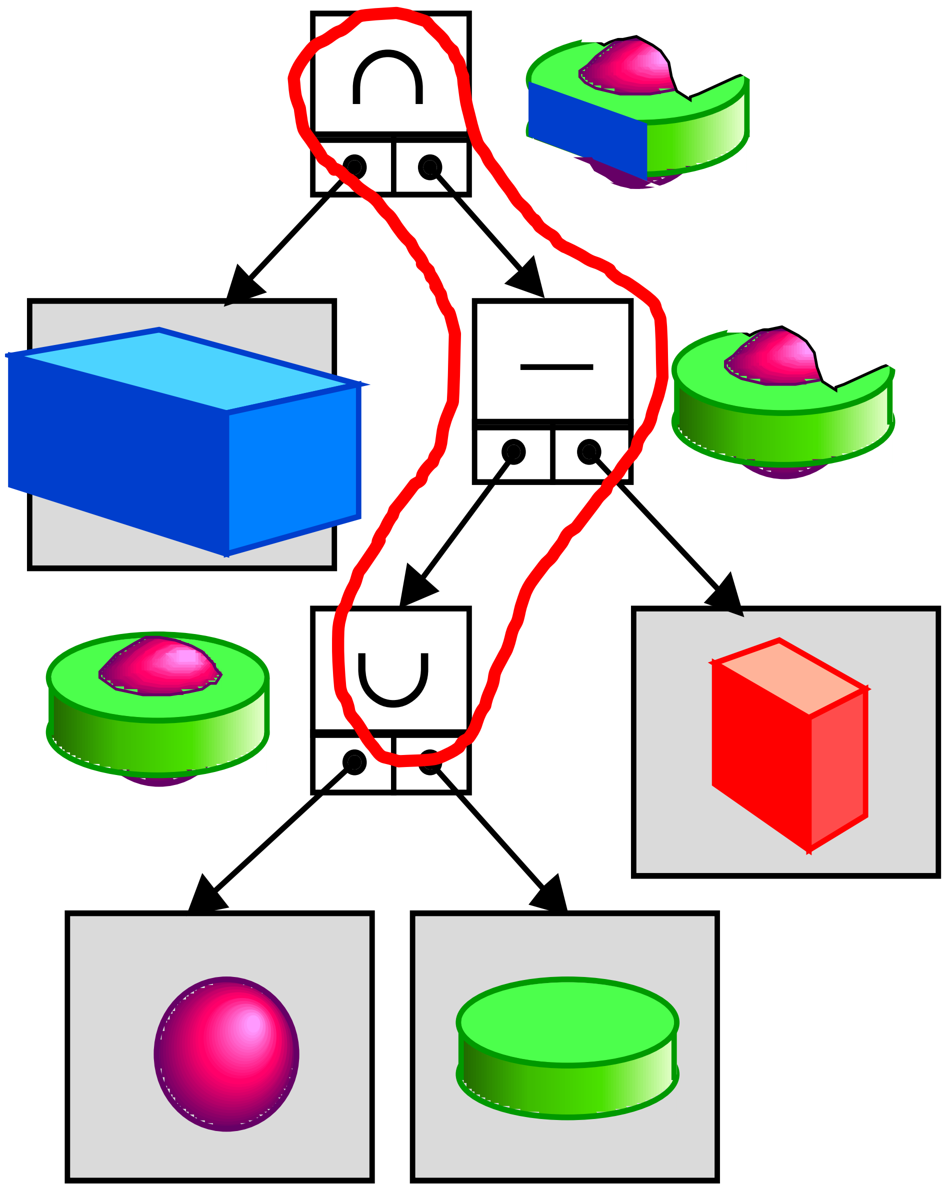
Objektrepräsentation

- Bei einer B-Rep wird nur die Oberfläche der Objekte beschrieben.
  ☒ **wahr** ☐ **falsch**
- CSG-Objekte werden durch die Operatoren Vereinigung, Durchschnitt und Mengendifferenz beschrieben. ☐ **wahr** ☐ **falsch**
- Der einzige Weg um CSG-Objekte zu zeichnen ist sie in BRep-Objekte umzurechnen. ☐ **wahr** ☐ **falsch**
- Ein Szenengraph ist eine genormte Datenstruktur zum Austausch geometrischer Daten. ☐ **wahr** ☐ **falsch**

CSG Data Structure

Every object is assembled from simple solids with

set operations



data structure: **binary tree**

recursive evaluation

Objektrepräsentation

- Bei einer B-Rep wird nur die Oberfläche der Objekte beschrieben.
  ☒ **wahr** ☐ **falsch**

- CSG-Objekte werden durch die Operatoren Vereinigung, Durchschnitt und Mengendifferenz beschrieben. ☒ **wahr** ☐ **falsch**

- Der einzige Weg um CSG-Objekte zu zeichnen ist sie in BRep-Objekte umzurechnen. ☐ **wahr** ☐ **falsch**

- Ein Szenengraph ist eine genormte Datenstruktur zum Austausch geometrischer Daten. ☐ **wahr** ☐ **falsch**

Rendering of CSG Trees

- transform into B-Rep and use normal hidden surface algorithm

  or

- render directly with ray casting (or with ray tracing)

Objektrepräsentation

- Bei einer B-Rep wird nur die Oberfläche der Objekte beschrieben.
  ☐ **wahr** ☐ **falsch**
- CSG-Objekte werden durch die Operatoren Vereinigung, Durch-schnitt und Mengendifferenz beschrieben. ☐ **wahr** ☐ **falsch**
- Der einzige Weg um CSG-Objekte zu zeichnen ist sie in BRep-Objekte umzurechnen. ☐ **wahr** ☐ **falsch**
- Ein Szenengraph ist eine genormte Datenstruktur zum Austausch geometrischer Daten. ☐ **wahr** ☐ **falsch**

Scene Graphs

- object-oriented data structure
  - directed acyclic graph
- describes logical and/or spatial relationship of scene objects
- describes groups of (groups of …) objects
- no exact definition
- used in most graphics systems, e.g.
  - OpenSceneGraph
  - VRML
  - X3D …

Objektrepräsentation

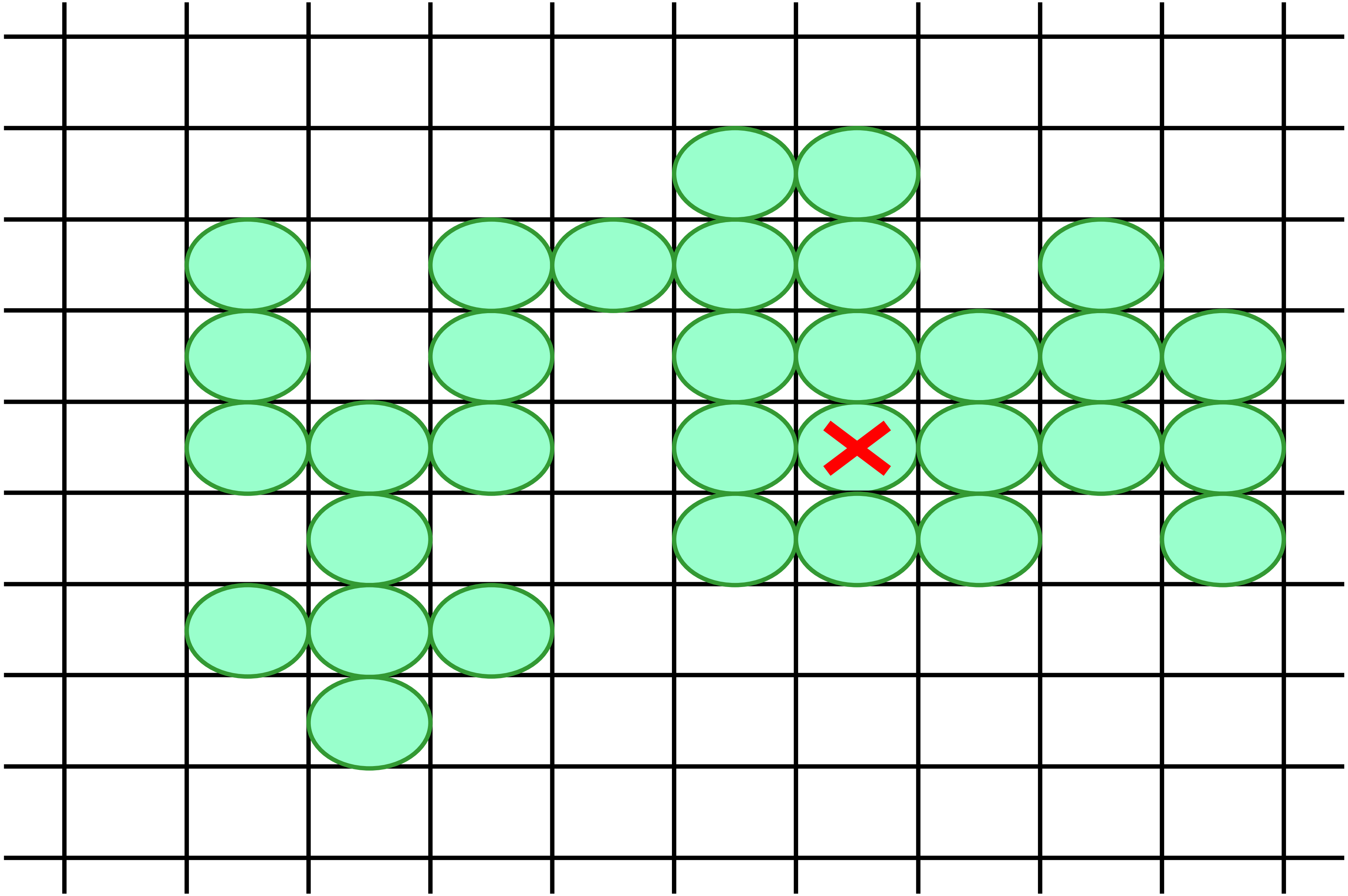
- Bei einer B-Rep wird nur die Oberfläche der Objekte beschrieben.
  ☒ wahr ☐ falsch

- CSG-Objekte werden durch die Operatoren Vereinigung, Durchschnitt und Mengendifferenz beschrieben. **☒ wahr ☐ falsch**

- Der einzige Weg um CSG-Objekte zu zeichnen ist sie in BRep-Objekte umzurechnen. **☐ wahr ☒ falsch**

- Ein Szenengraph ist eine genormte Datenstruktur zum Austausch geometrischer Daten. **☐ wahr ☒ falsch**

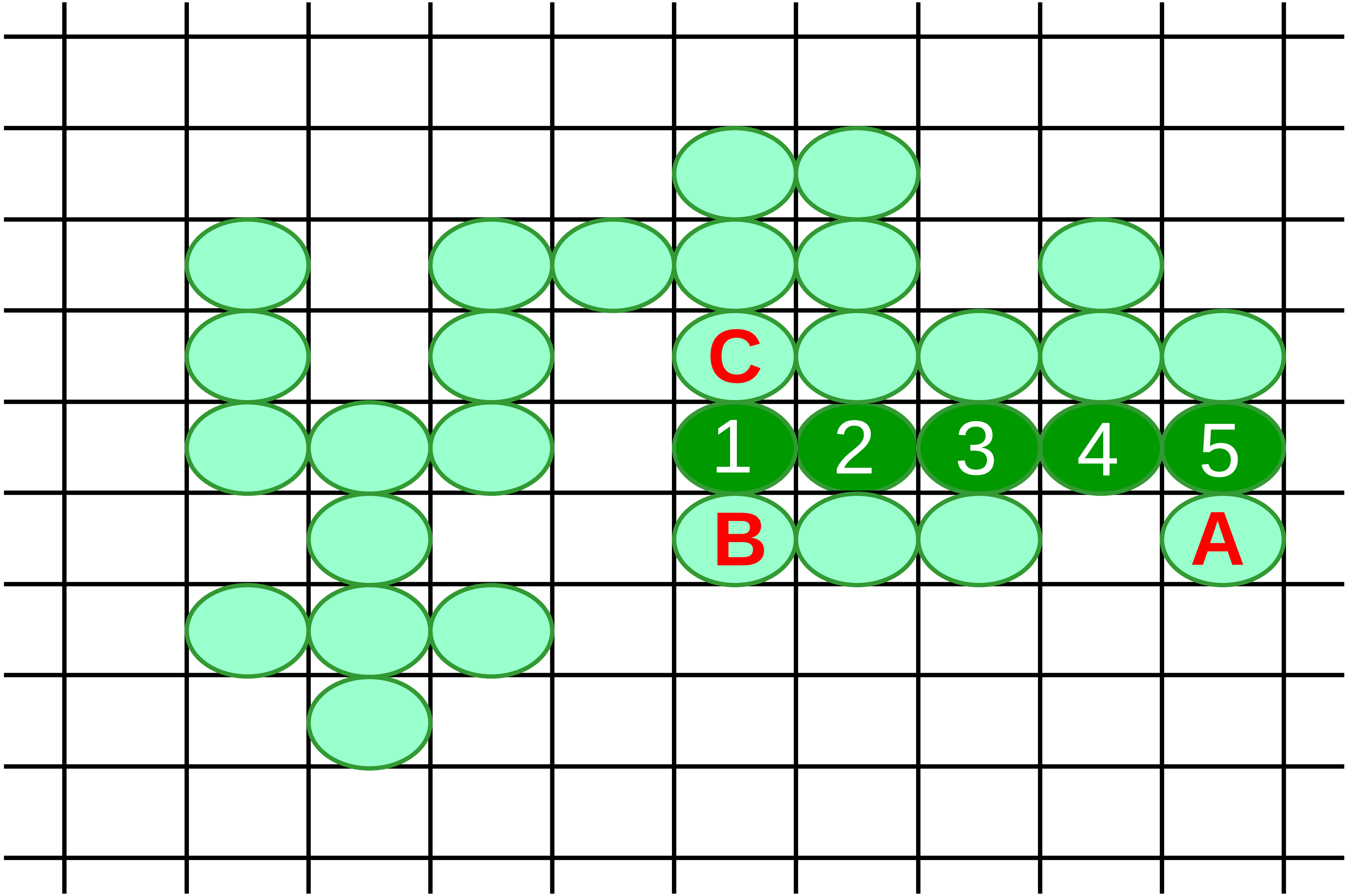
Span Flood-Fill



Stack:

x

Span Flood-Fill



Stack:

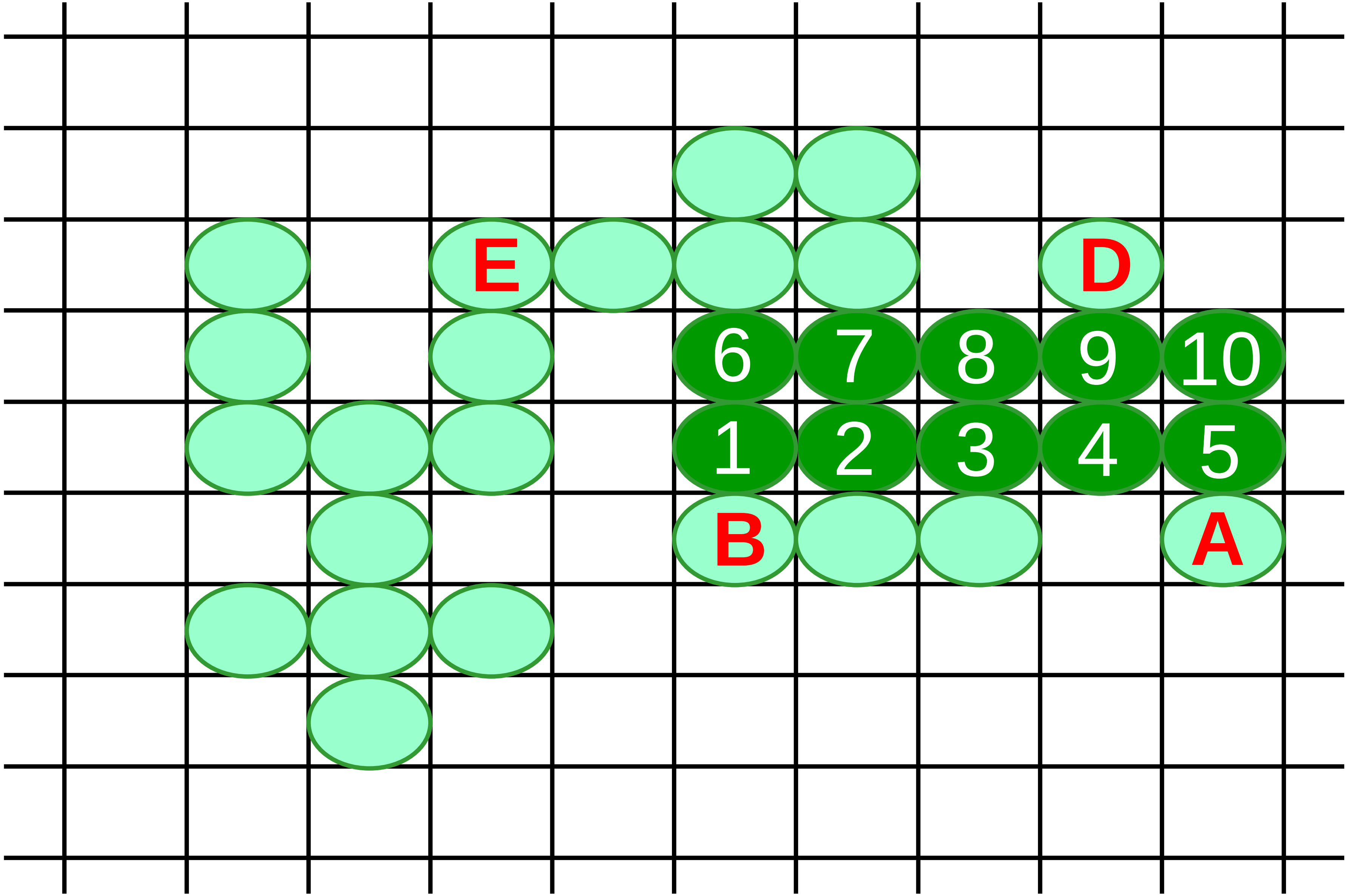
~~x~~

A

B

C

Span Flood-Fill



Stack:

A

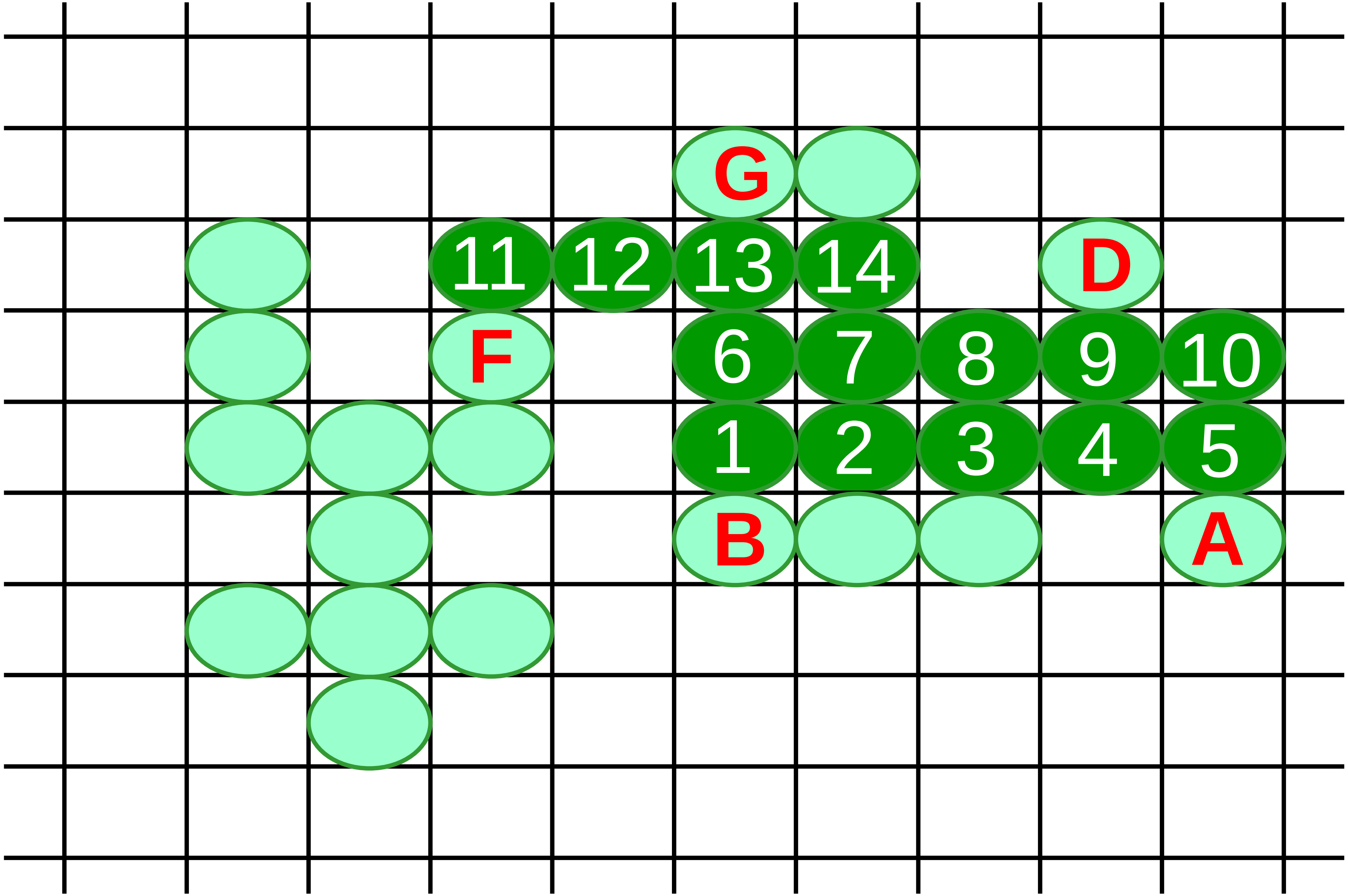
B

~~C~~

D

E

Span Flood-Fill



Stack:

A

B

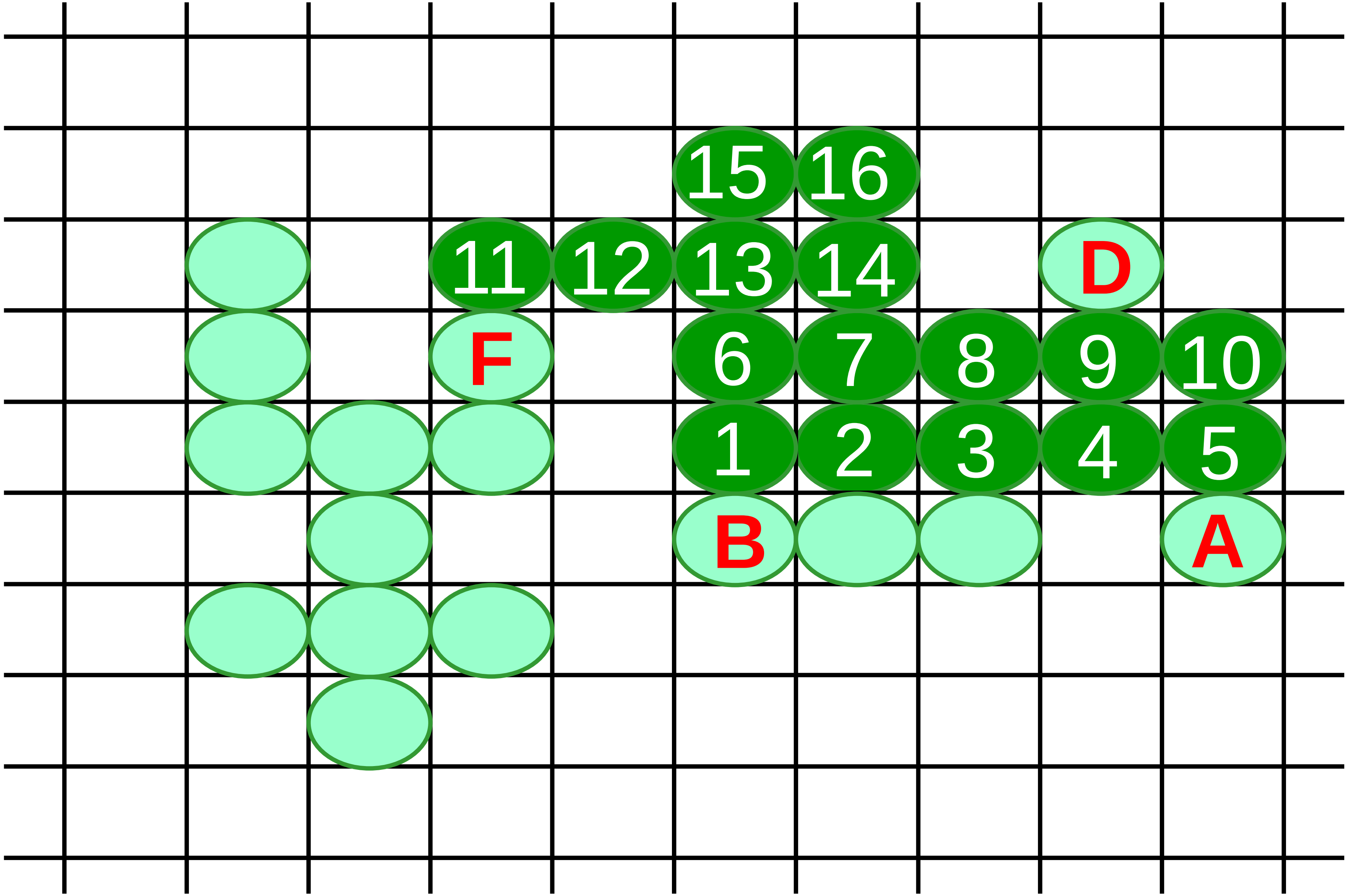
D

~~E~~

F

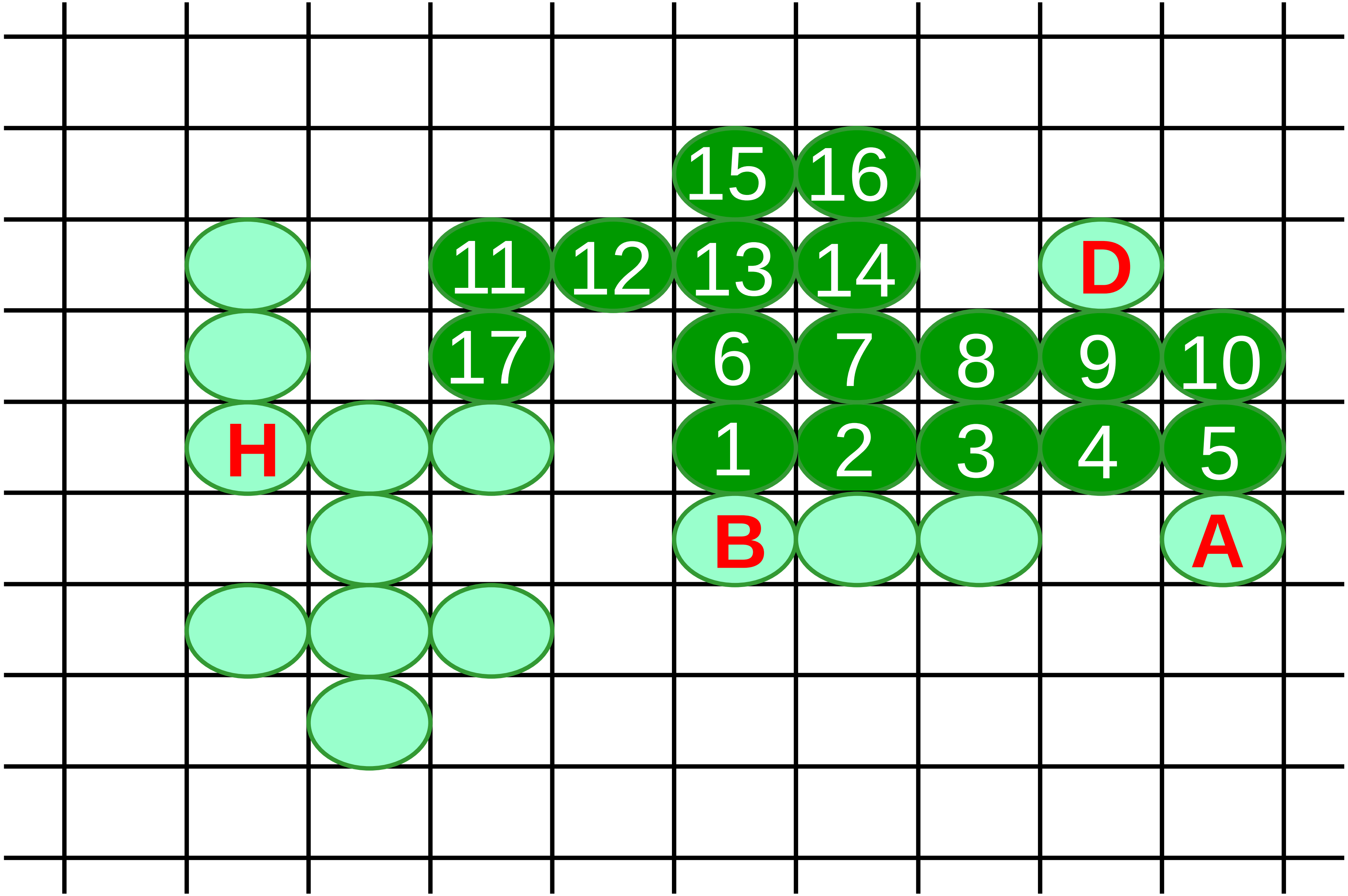
G

Span Flood-Fill



Stack:
A
B
D
F
~~G~~

Span Flood-Fill



Stack:

A

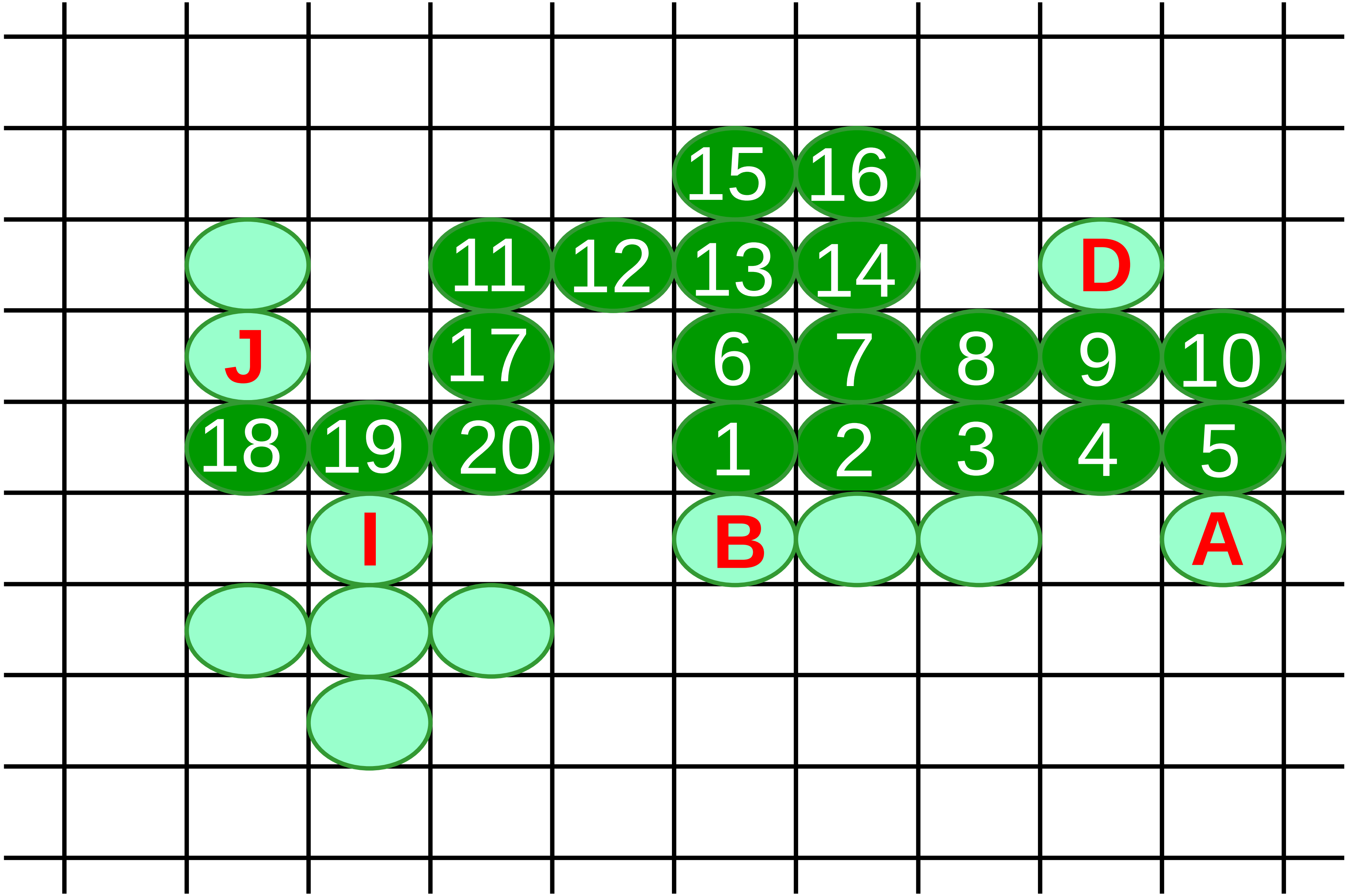
B

D

~~F~~

H

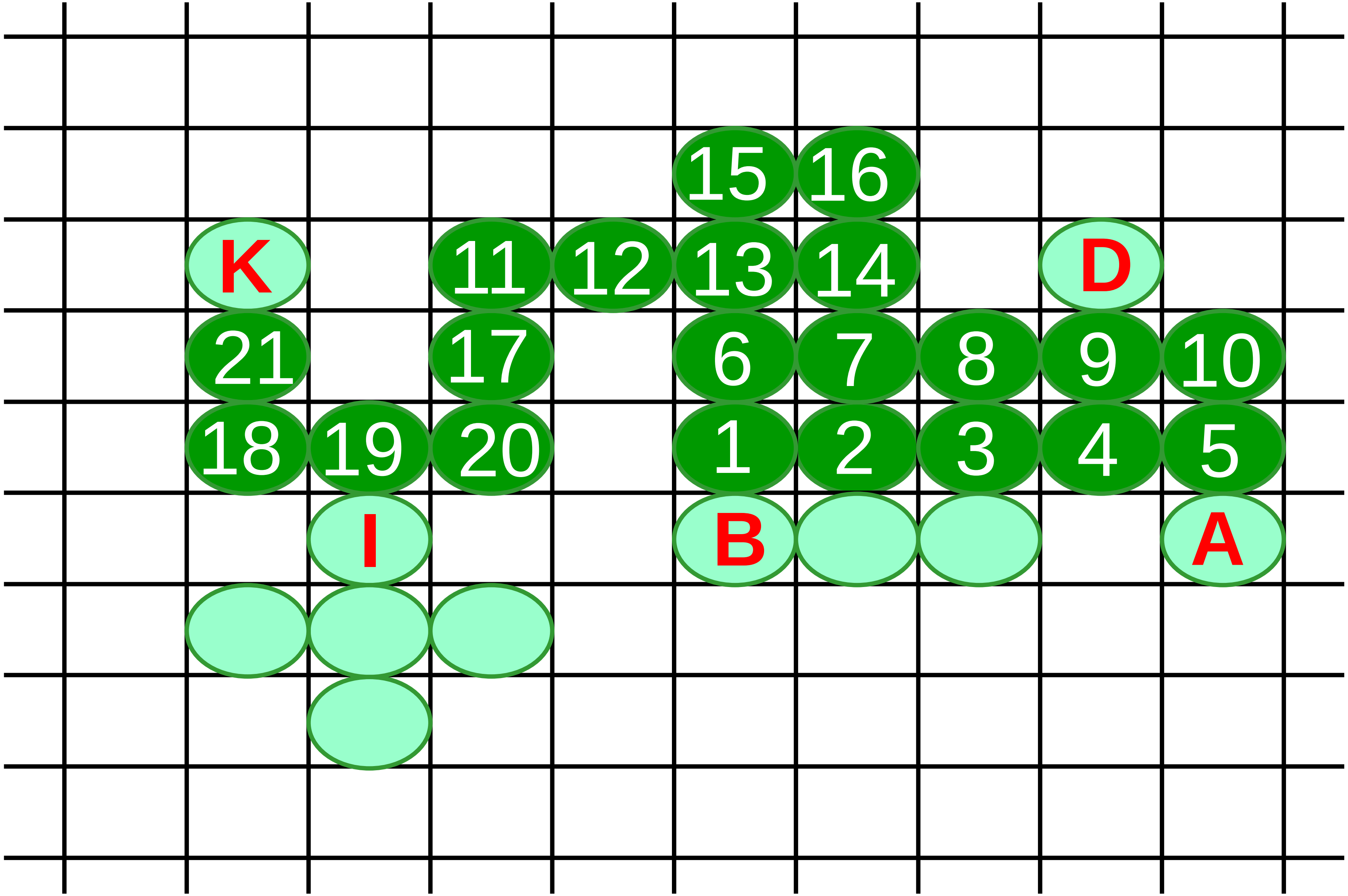
Span Flood-Fill



Stack:

A
B
D
~~H~~
I
J

Span Flood-Fill



Stack:

A

B

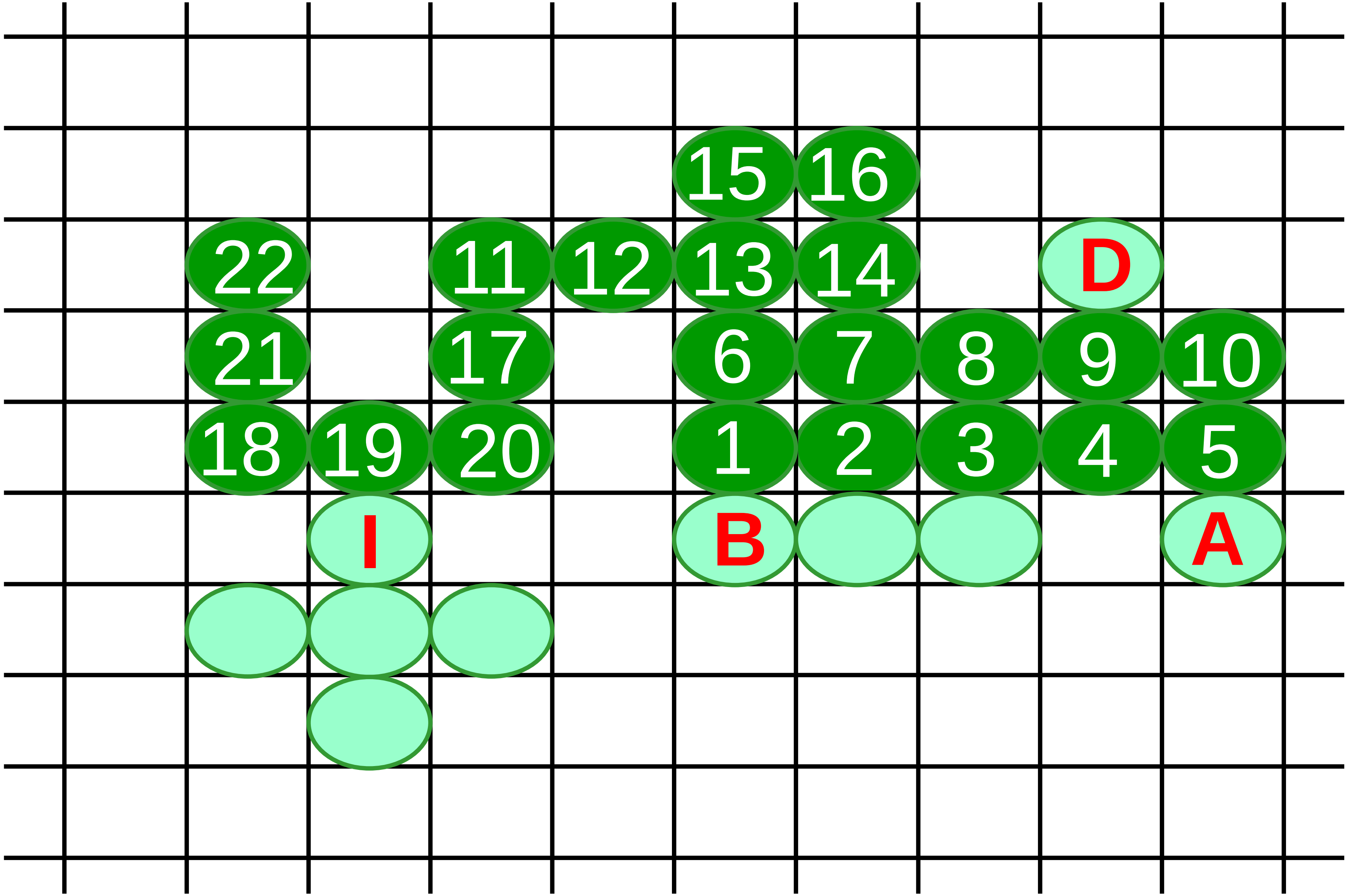
D

I

~~J~~

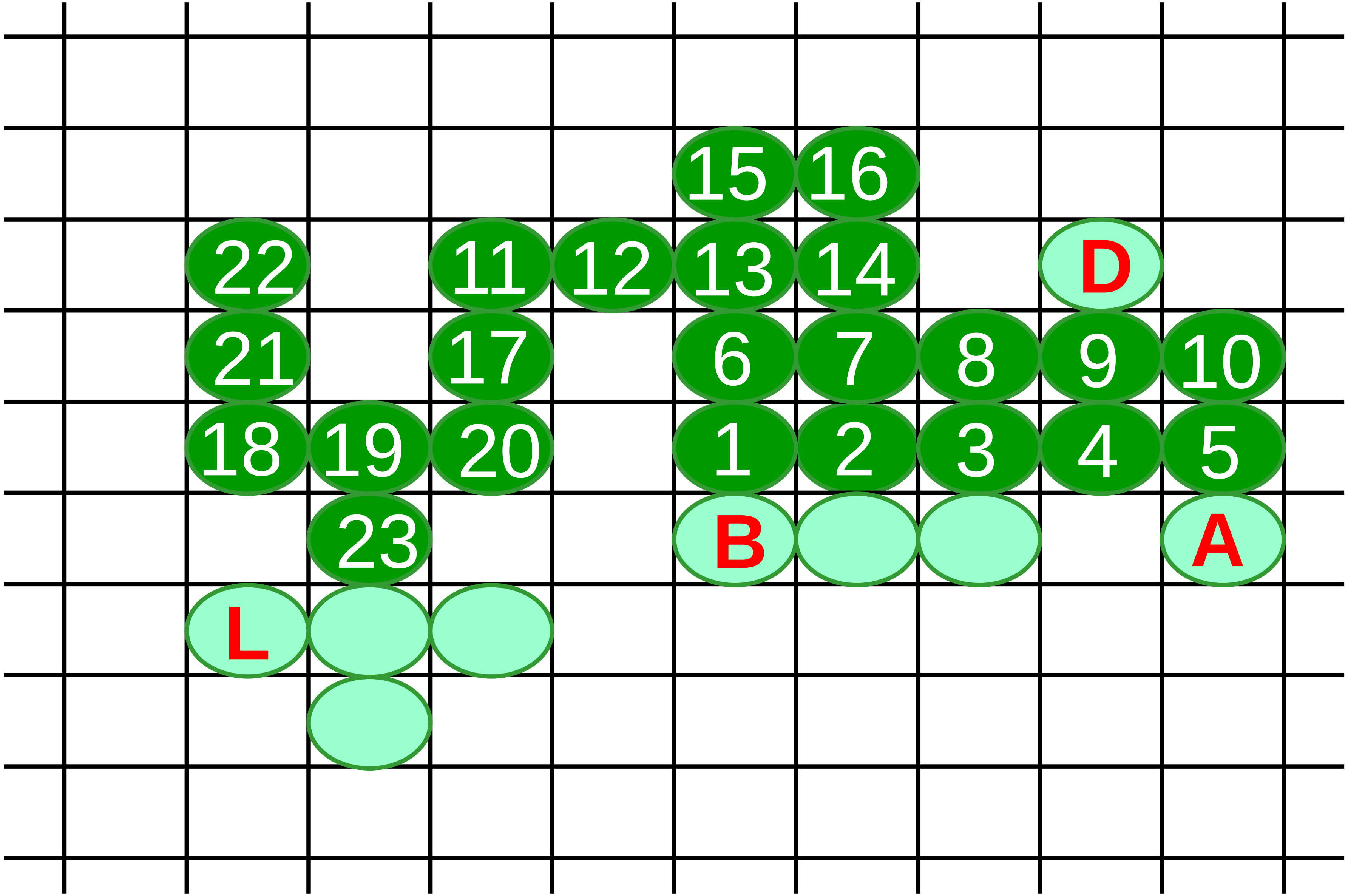
K

Span Flood-Fill



Stack:
A
B
D
I
~~K~~

Span Flood-Fill



Stack:

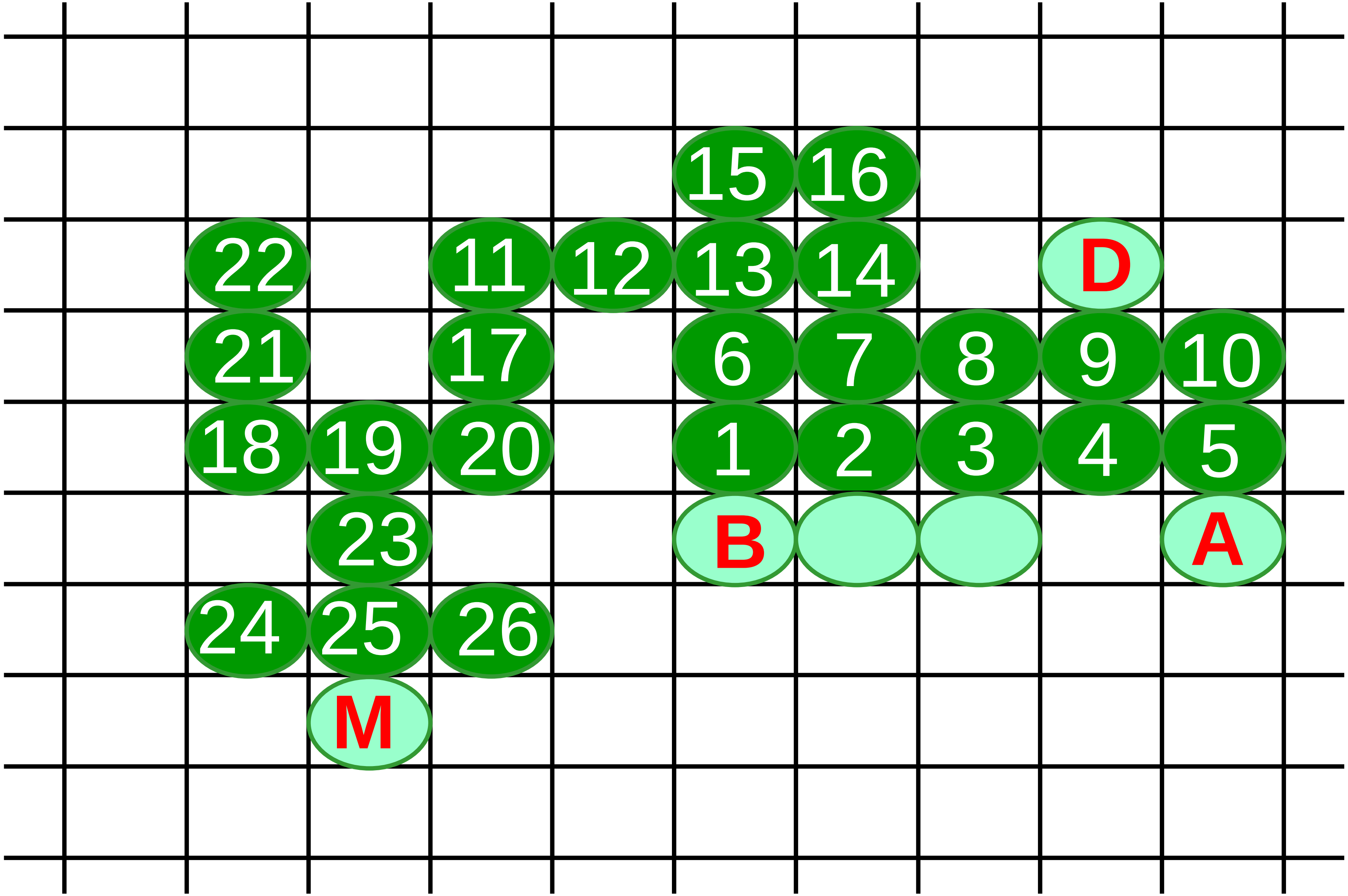
A

B

D

L

Span Flood-Fill



Stack:

A

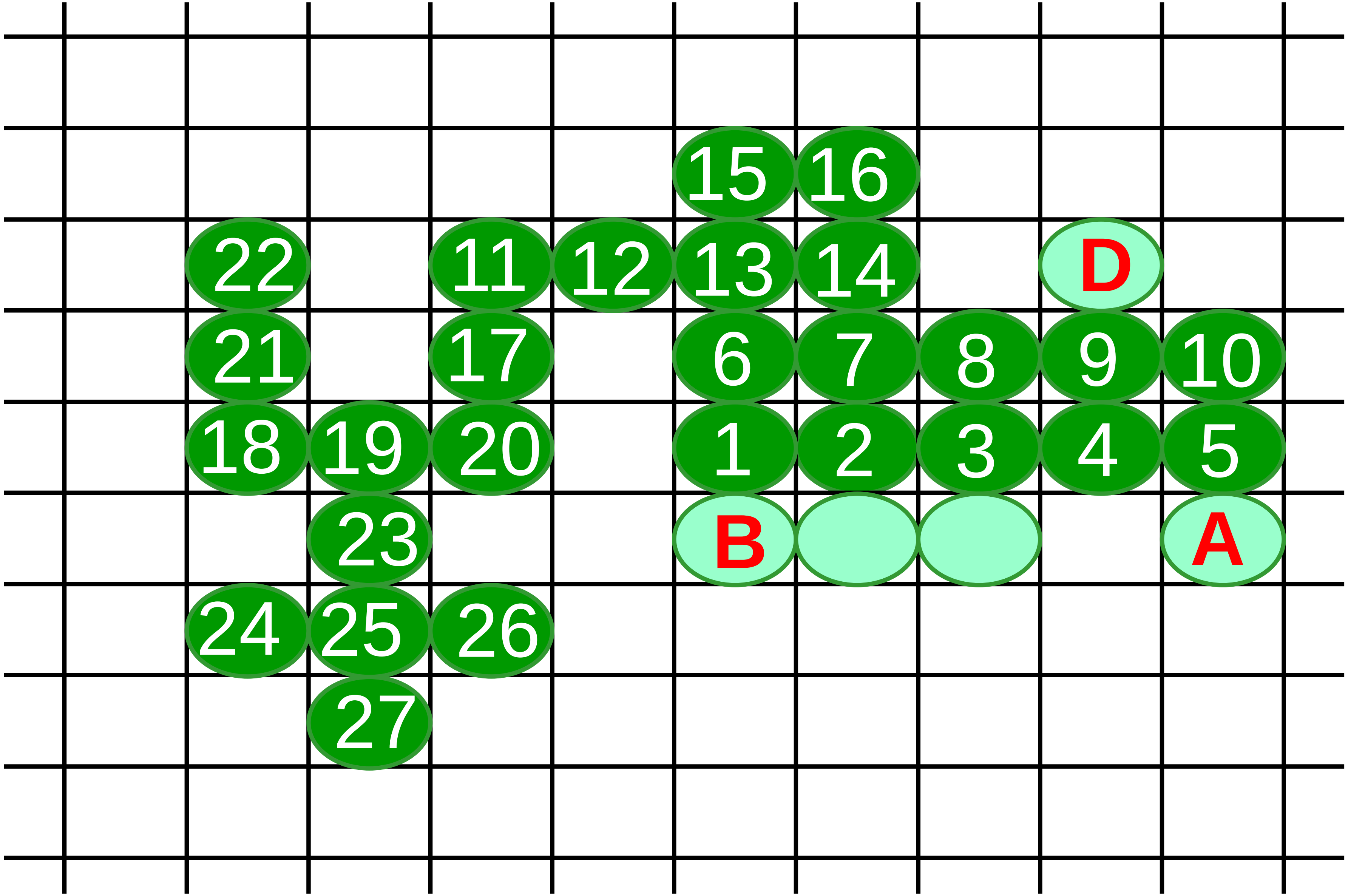
B

D

~~L~~

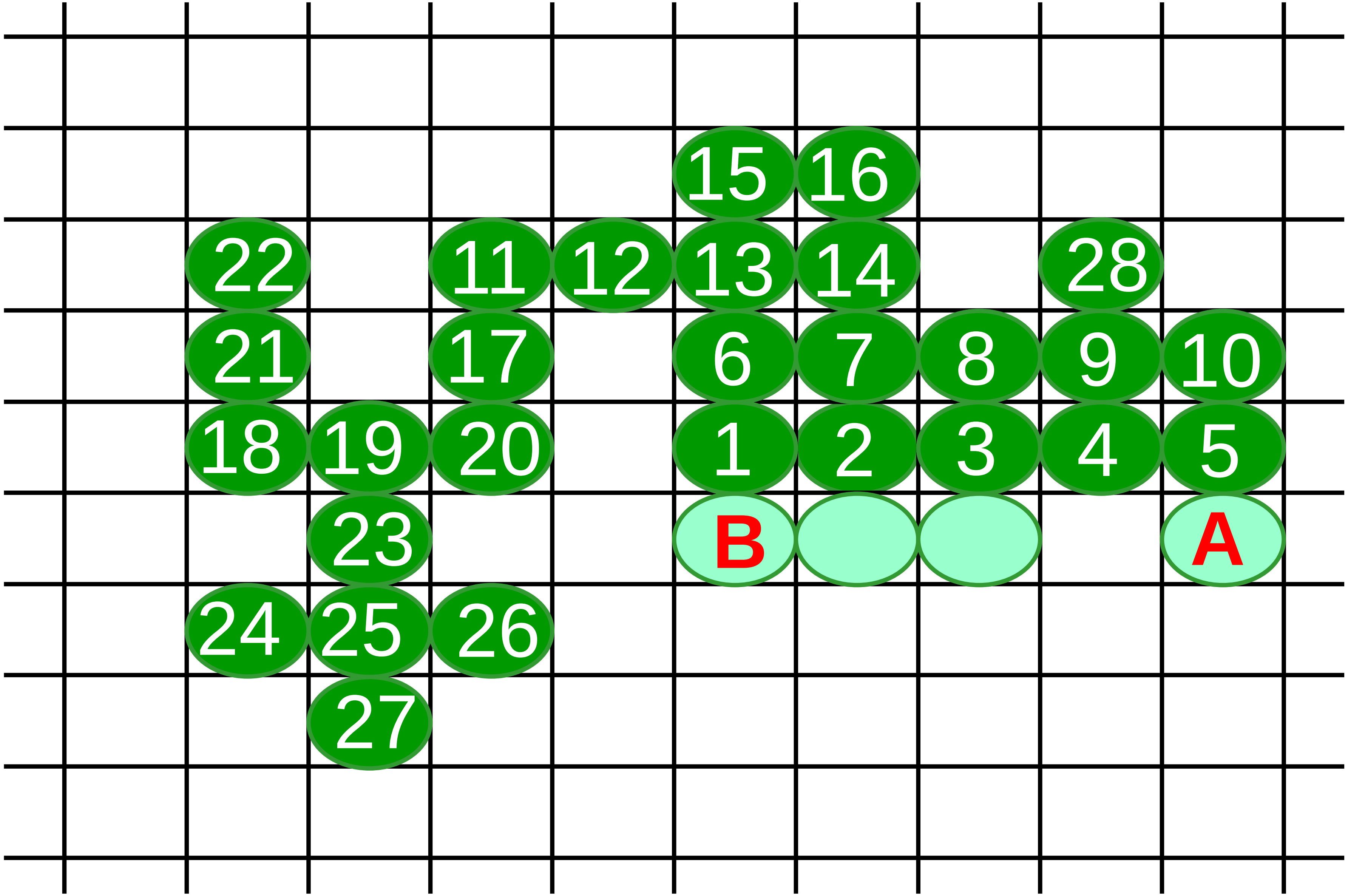
M

Span Flood-Fill



Stack:
A
B
D
~~M~~

Span Flood-Fill

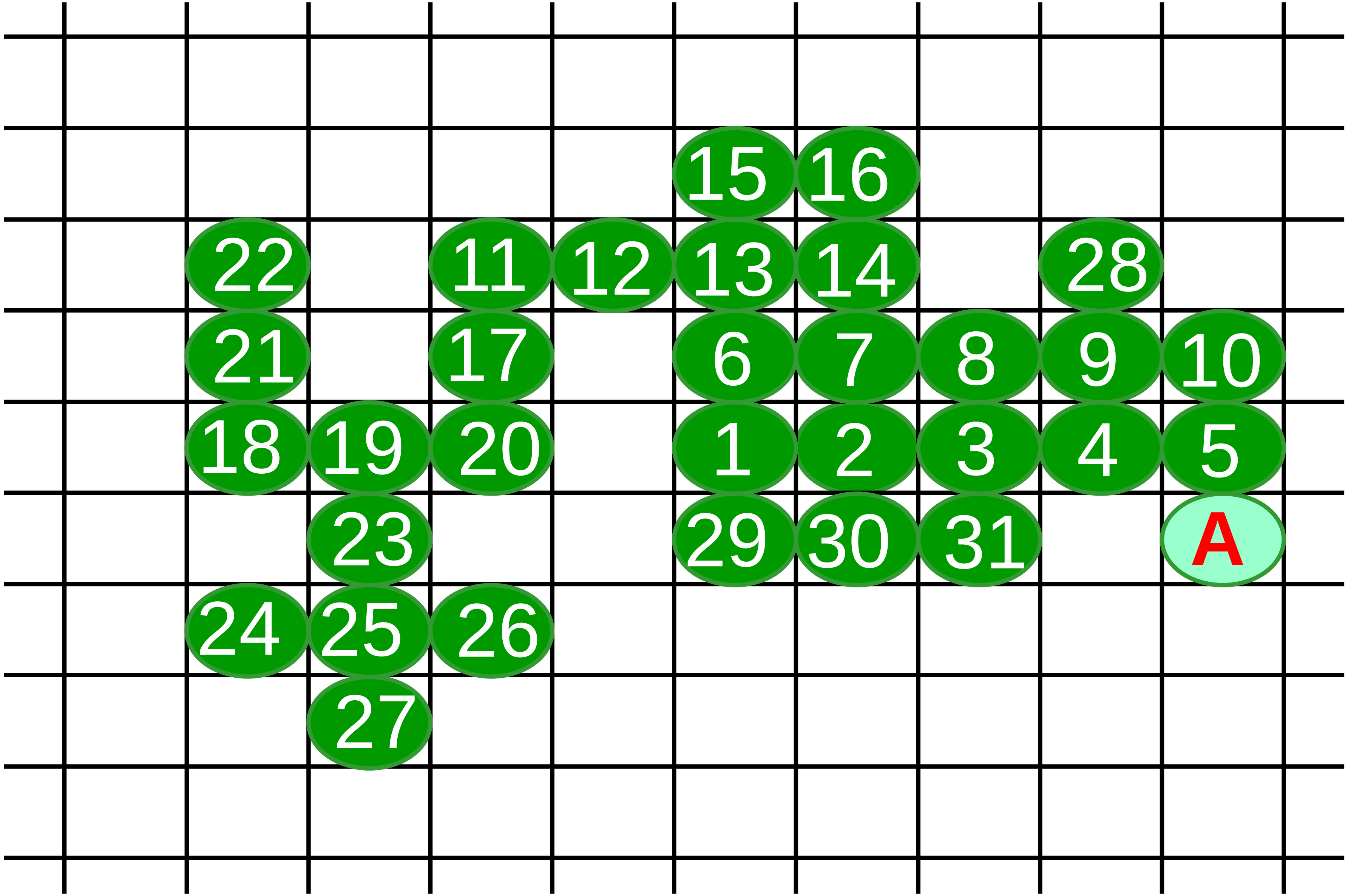


Stack:

A

B

~~D~~

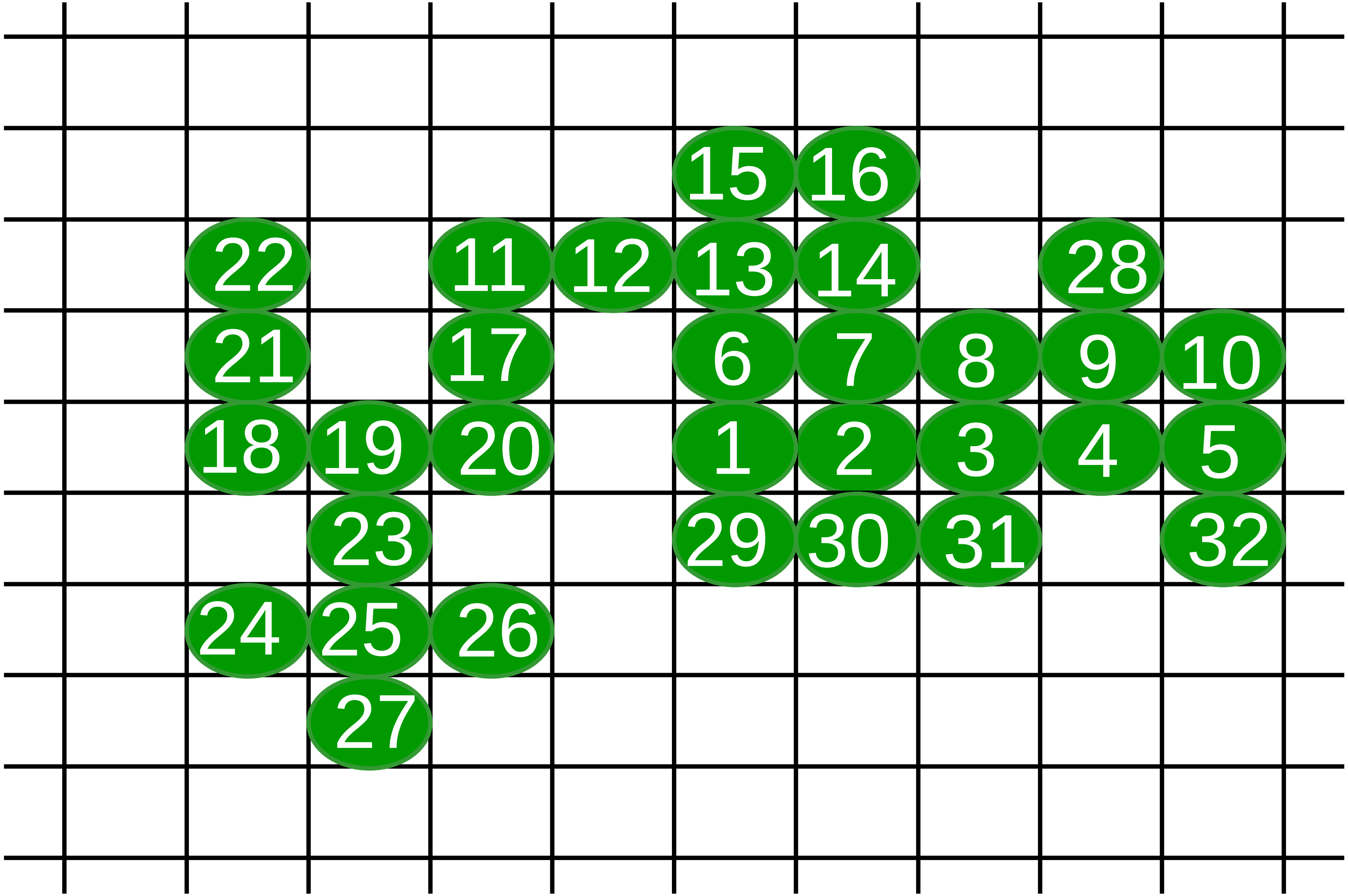

15
16
22
11
12
13
14
28
21
17
6
7
8
9
10
18
19
20
1
2
3
4
5
23
29
30
31
A
24
25
26
27

Stack:

A

~~B~~

Span Flood-Fill



Stack:

~~A~~

fertig!

Basiswissen Matrizen (1)

Multiplikation eines nD-Vektors mit einer n×n-Matrix bewirkt eine ***affine Transformation***:

Kollinearität, Parallelität, Teilverhältnisse bleiben erhalten.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \qquad P = \begin{bmatrix} p_1 \\ p_2 \\ \dots \\ p_n \end{bmatrix} \qquad P' = A \cdot P$$

Die ***Inverse*** A^{-1} einer Matrix A bewirkt: $P = A^{-1} \cdot P'$

Außerdem gilt natürlich $A^{-1} \cdot A = I$, wobei $I = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & & \ddots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}$
die ***Einheitsmatrix*** ist.

Basiswissen Matrizen (2)

Die ***Addition*** von Matrizen erfolgt elementweise:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & & & \dots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} = \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \dots & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{22} & \dots & a_{2n}+b_{2n} \\ \dots & \dots & & \\ a_{n1}+b_{n1} & a_{n2}+b_{n2} & \dots & a_{nn}+b_{nn} \end{bmatrix}$$

Die ***Multiplikation*** von Matrizen erfolgt durch skalare Multiplikation von Zeilen- und Spaltenvektoren:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \bullet \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & & & \dots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & a_{21} \cdot b_{12} + \dots + a_{2n} \cdot b_{n2} & \dots & c_{2n} \\ \dots & \dots & & \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{bmatrix}$$

Basiswissen Matrizen (3)

Für **Matrizen-Addition** gelten:

Assoziativgesetz: $(A+B)+C = A+(B+C)$

Kommutativgesetz: $A+B = B+A$

Für **Matrizen-Multiplikation** gelten:

Assoziativgesetz: $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

Kommutativgesetz ***NICHT***: $A \cdot B \neq B \cdot A$

[Weiters gilt das:

Distributivgesetz: $(A+B) \cdot C = A \cdot C + B \cdot C$ und $A \cdot (B+C) = A \cdot B + A \cdot C$]

Die ***Transponierte*** $\boldsymbol{A^T}$ einer Matrix A erhält man durch Spiegelung aller Elemente an der Hauptachse:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \qquad A^T = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \dots & & & \dots \\ a_{1n} & a_{2n} & \dots & a_{nn} \end{bmatrix}$$

Eine ***Matritze*** gibt es nicht! Ähnliche Begriffe:

Matrize (Druckvorlage)

Matratze

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & & & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

Matrix

Beispiel Transformation

Ein Turm mit dem Mittelpunkt seiner Grundfläche an der Stelle (0,0,0) soll in der Höhe um den Faktor 3 vergrößert werden, danach um seine senkrechte Achse um den Winkel 60° gegen den Uhrzeigersinn rotiert werden und schließlich um 20 (Meter) vom Betrachter weg geschoben werden. Der Betrachter schaut wie üblich in –z-Richtung waagrecht zum Turm, auch die x-Achse ist horizontal. *Berechnen Sie die Transformationsmatrix!*

x-Achse und z-Achse waagrecht, daher y-Achse senkrecht

T_1: Skalierung um 3 in y-Richtung

T_2: Drehung um 60° um y-Achse in math. positive Richtung

T_3: Verschieben in negative z-Richtung um 20

Beispiel Transformation (2)

Skalierung (1,3,1), y-Rotation +60°, Translation (0,0,-20)

1. $M_1 = S(1,3,1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

2. $M_2 = R_y(+60°) = \begin{bmatrix} 1/2 & 0 & \sqrt{3}/2 & 0 \\ 0 & 1 & 0 & 0 \\ -\sqrt{3}/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3. $M_3 = T(0,0,-20) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -20 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

$$\boxed{\mathbf{M} = \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1}$$

Beispiel Transformation (3)

Skalierung (1,3,1), y-Rotation +60°, Translation (0,0,-20)

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -20 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{2} & 0 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{\sqrt{3}}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{1}{2} & 0 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{\sqrt{3}}{2} & 0 & \frac{1}{2} & -20 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{1}{2} & 0 & \frac{\sqrt{3}}{2} & 0 \\ 0 & 3 & 0 & 0 \\ -\frac{\sqrt{3}}{2} & 0 & \frac{1}{2} & -20 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boxed{\mathbf{M} = \mathbf{M}_3 \cdot \mathbf{M}_2 \cdot \mathbf{M}_1}$$

Reminder: Product of Vectors

$$V_1 = \begin{pmatrix} a_1 \\ b_1 \\ c_1 \end{pmatrix} \qquad V_2 = \begin{pmatrix} a_2 \\ b_2 \\ c_2 \end{pmatrix}$$

- ***scalar product:***

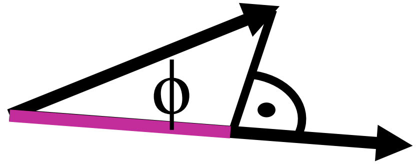
$$V_1 \cdot V_2 = ?$$

- ***cross product (vector product):***

$$V_1 \times V_2 = ?$$

- ***scalar product:***

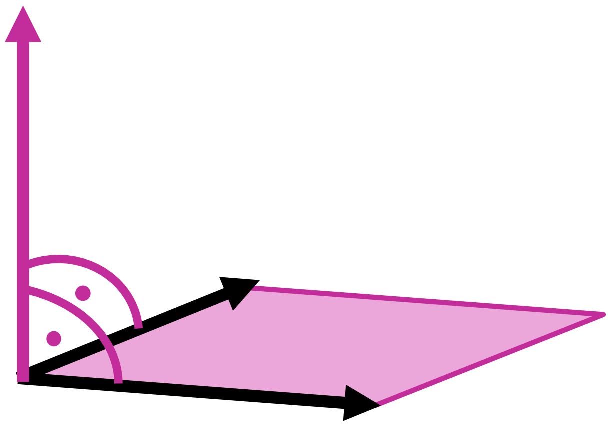
$$V_1 \cdot V_2 = \begin{pmatrix} a_1 \\ b_1 \\ c_1 \end{pmatrix} \cdot \begin{pmatrix} a_2 \\ b_2 \\ c_2 \end{pmatrix} = a_1 a_2 + b_1 b_2 + c_1 c_2$$



$$V_1 \cdot V_2 = |V_1||V_2|\cos\phi$$

- ***cross product (vector product):***

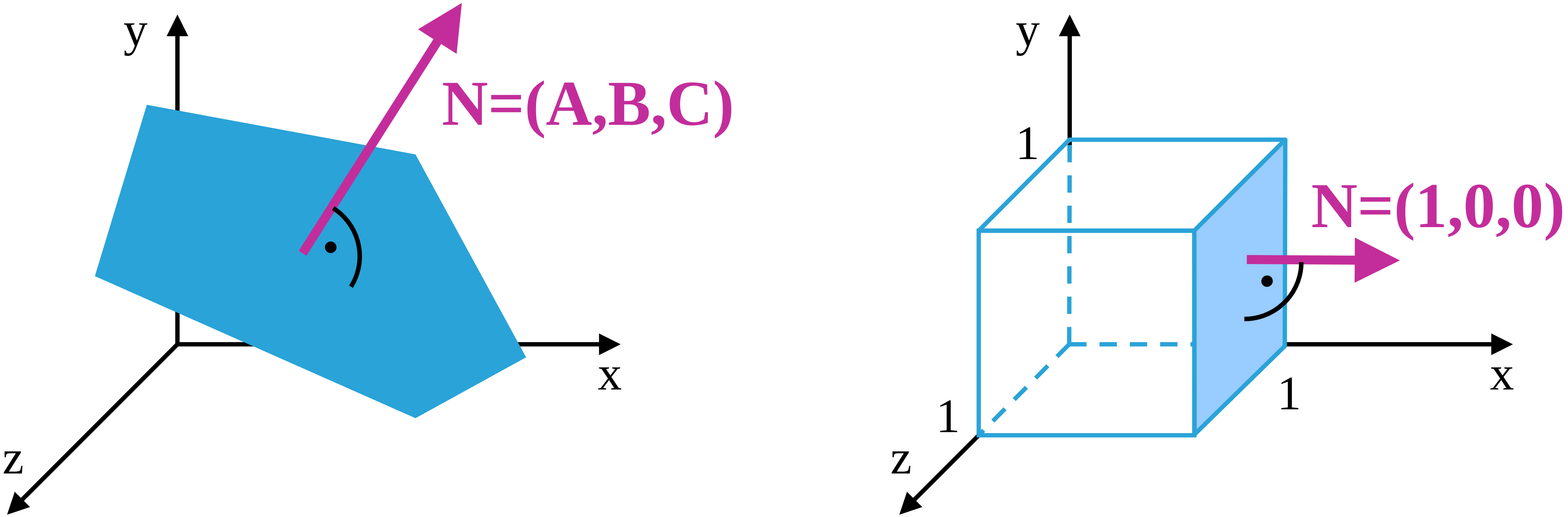
$$V_1 \times V_2 = \begin{pmatrix} a_1 \\ b_1 \\ c_1 \end{pmatrix} \times \begin{pmatrix} a_2 \\ b_2 \\ c_2 \end{pmatrix} = \begin{pmatrix} b_1 c_2 - c_1 b_2 \\ c_1 a_2 - a_1 c_2 \\ a_1 b_2 - b_1 a_2 \end{pmatrix}$$



$$|V_1 \times V_2| = |V_1||V_2|\sin\phi$$

$$\boxed{Ax + By + Cz + D = 0}$$

- plane parameters **A,B,C,D**
- normal **(A,B,C)**



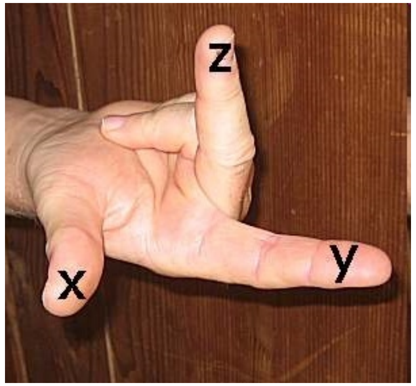
$Ax + By + Cz + D = 0$ for points on the surface

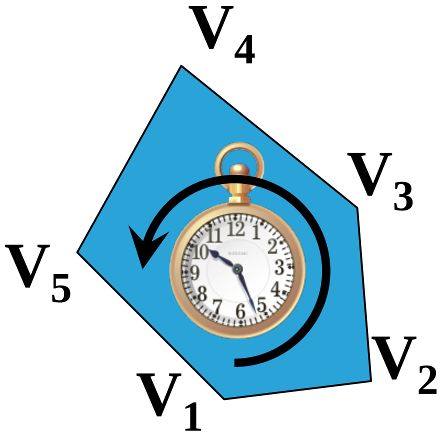
< 0 for points behind

> 0 for points in front

if (1) right-handed coordinate system

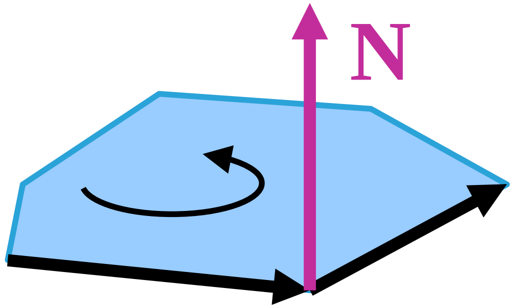
(2) polygon points are ordered counterclockwise





$\mathbf{V_1}, \mathbf{V_2}, \mathbf{V_3}$ counterclockwise $\Rightarrow$

normal vector $\mathbf{N} = (\mathbf{V_2} - \mathbf{V_1}) \times (\mathbf{V_3} - \mathbf{V_2})$

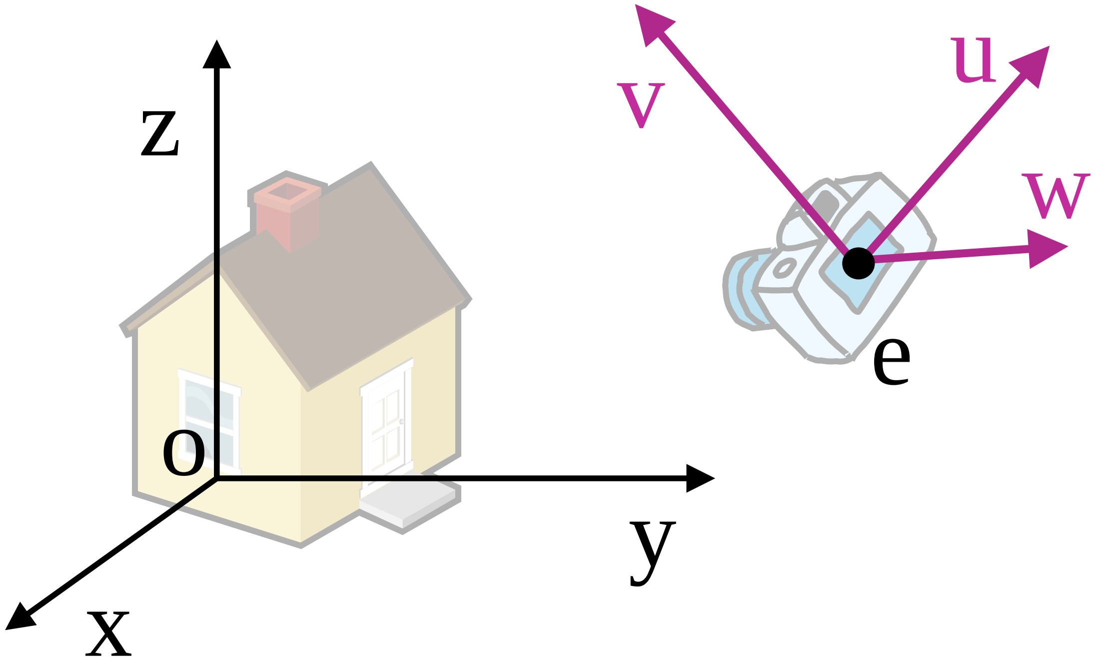


Beispiel Kamerakoordinaten

Eine Kamera liegt im Punkt (3,4,5) und blickt auf den Punkt (3,2,1). Geben Sie die normalisierten Achsen des Kamerakoordinatensystems an, wenn die senkrechte Achse der Weltkoordinaten nach oben (z) zeigt.

Viewing: Camera Transformation (1)

- view reference point
  - origin of camera coordinate system
  - gaze direction or look-at point



right-handed camera-coordinate system,
with axes u, v, w, relative to world-coordinate scene

Viewing: Camera Transformation (2)

e … eye position

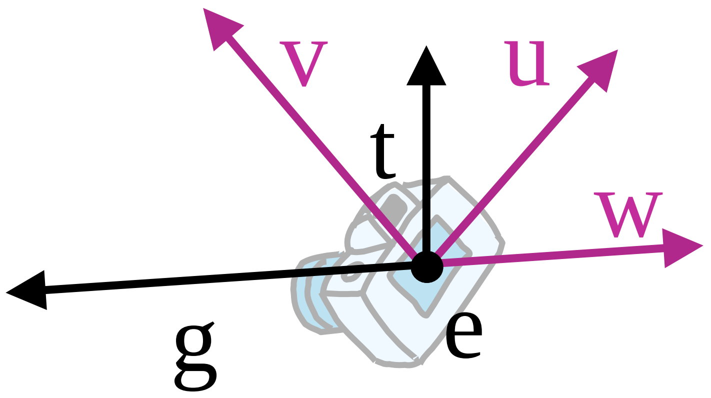
g … gaze direction (positive w-axis points to the viewer)

t … view-up vector

$$w = -\frac{g}{|g|}$$

$$u = \frac{t \times w}{|t \times w|}$$

$$v = w \times u$$



Beispiel Kamerakoordinaten

Eine Kamera liegt im Punkt (3,5,5) und blickt auf den Punkt (3,2,1). Geben Sie die normalisierten Achsen des Kamerakoordinatensystems an, wenn die senkrechte Achse der Weltkoordinaten nach oben (z) zeigt.

$$g = (3, 2, 1) - (3, 5, 5) = (0, -3, -4) \qquad |g| = \sqrt{0^2+3^2+4^2} = 5$$

$$w = -g / |g| = (0, 3, 4) / 5 = \underline{(0, 3/5, 4/5)}$$

$$t = (0, 0, 1)$$

$$t \times w = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 3/5 \\ 4/5 \end{pmatrix} = \begin{pmatrix} -3/5 \\ 0 \\ 0 \end{pmatrix}$$

$$u = t \times w / |t \times w| = (-3/5, 0, 0) / 3/5 = \underline{(-1, 0, 0)}$$

$$v = w \times u = \begin{pmatrix} 0 \\ 3/5 \\ 4/5 \end{pmatrix} \times \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ -4/5 \\ 3/5 \end{pmatrix} = \underline{(0, -4/5, 3/5)}$$